TPC Benchmark™ C
Full Disclosure Report
for
IBM Netfinity 8500R
using
Microsoft SQL Server 7.0 Enterprise Edition
and
Windows NT Server 4.0 Enterprise Edition

Submitted for Review
February 25, 2000

First Edition - February 2000

Trademarks

IBM is a registered trademark and Netfinity is a trademark of International Business Machines Corporation.

The following terms used in this publication are trademarks of other companies as follows: TPC Benchmark, trademark of Transaction Processing Performance Council; Intel, Pentium and Xeon are registered trademarks of Intel Corporation; Microsoft, Windows 2000, Windows NT and BenchCraft are trademarks or registered trademarks of Microsoft Corporation. Other company, product, or service names, which may be denoted by two asterisks (**), may be trademarks or service marks of others.

Notes

[1] MHz only measures microprocessor internal clock speed, not application performance. Many factors affect application performance.
[2] When referring to hard disk capacity, GB, or gigabyte, means one thousand million bytes. Total user-accessible capacity depends on operating environment.

Abstract

IBM Corporation conducted the TPC Benchmark™ C on the IBM Netfinity* 8500R configured as a client/server system. This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification, Revision 3.5, including the methodology used to achieve the reported results. All testing fully complied with this revision level.

The software used on the IBM Netfinity 8500R system includes Microsoft** Windows** NT Server 4.0, Enterprise Edition, operating system and Microsoft SQL Server 7.0, Enterprise Edition, database.

Two standard metrics, transactions per minute-C (tpmC) and price per tpmC ($/tpmC), are reported as required by the TPC Benchmark C Standard Specification.

The benchmark results are summarized in the following table.

Hardware	Software	Total System Cost	tpmC	$/tpmC	Total Solution Availability Date
IBM Netfinity 8500R	Microsoft SQL Server 7.0, Enterprise Edition Microsoft Windows NT 4.0, Enterprise Edition	$763,922	40,251.15	$18.98	Feb. 25, 2000

The results of the benchmark and test methodology used were audited by Francois Raab of Info Sizing, Inc. The auditor's attestation letter is contained in Section 9 of this report.

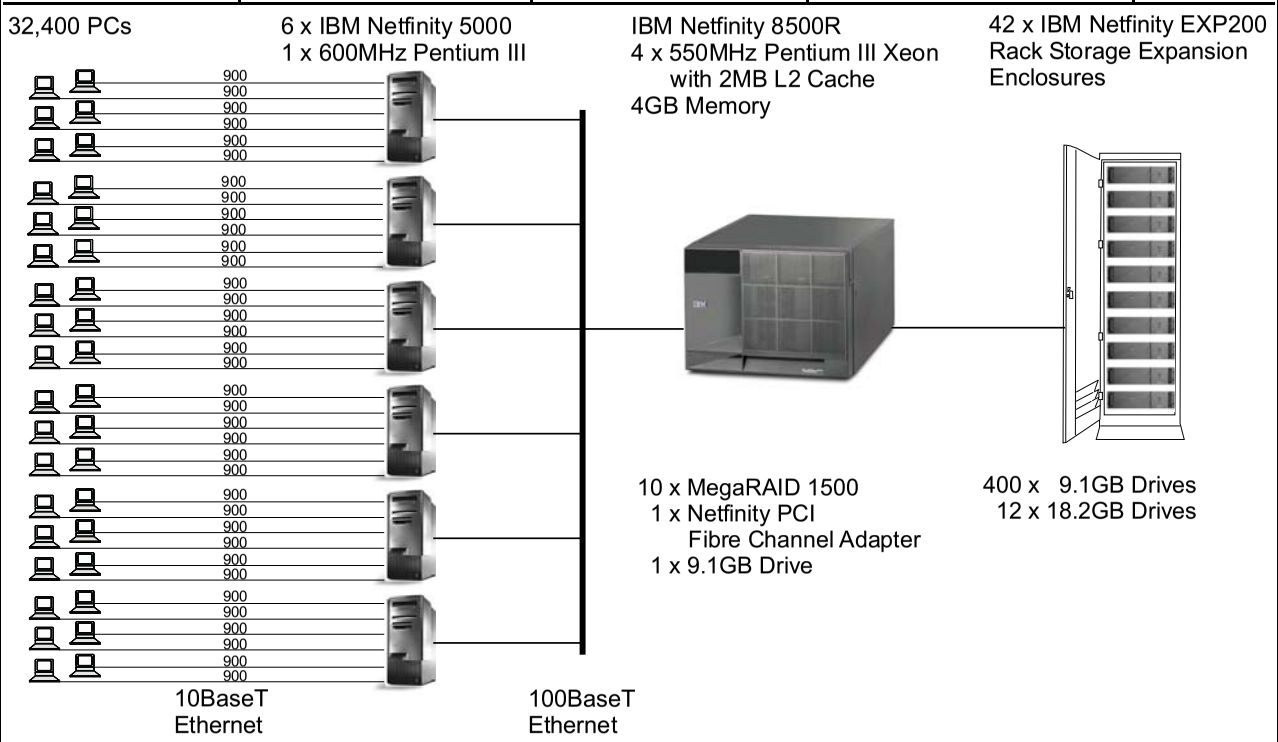
IBM Netfinity 8500R c/s

TPC-C Rev 3.5

Report Date: Feb. 25, 2000

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$763,922	**40,251.15 tpmC**	**$18.98/tpmC**	**Feb. 25, 2000**

Processors	Database Manager	Operating System	Other Software	Number of Users
8 Intel Pentium III Xeon 550MHz 2MB L2 Cache	**Microsoft SQL Server 7.0 Enterprise Edition**	**Microsoft Windows NT 4.0 Enterprise Edition**	**Microsoft Windows 2000 Server Microsoft Visual C++ 6.0**	**32,400**



System Component	Qty	Server	Qty	Each of Six Clients
Processors	**8**	**550MHz Pentium III Xeon w/2MB L2 Cache**	**1**	**600MHz Pentium III w/512KB L2 Cache**
Memory	**1**	**4096MB**	**1**	**512MB**
Disk Controllers	**10** **1**	**MegaRAID 1500-H** **Fibre Channel PCI**	**1**	**Wide Ultra SCSI Onboard**
Disk Drives	**401** **12**	**9.1GB (10000 rpm)** **18.2GB (10000 rpm)**	**1**	**9.1GB Hard Disk**
Total Storage		**3868GB**		
Tape Drive	**1**	**20/40GB SCSI Tape Drive**		

IBM Corporation	Netfinity 8500R c/s	TPC-C Revision 3.5
		Report Date: Feb. 25, 2000

Description	Order Number	Third-Party Brand	Pricing	Unit Price	Qty	Ext. Price	5-Yr. Maint.*
Server Hardware							
Netfinity 8500R 550MHz/2MB Pentium III Xeon	86816RY		1	$23,330	1	$23,330	$10,295
550MHz/2MB L2 Cache Processor Upgrade	33L5105		1	6,280	7	43,960	0
8500R Memory Expansion Card	28L4454		1	587	1	587	0
512MB ECC SDRAM RDIMM Memory Kit	20L0249		1	2,275	7	15,925	0
8500 >4-Way Enablement Kit (4X SRAM)	28L4727		1	1,990	1	1,990	0
AMI MegaRAID 1500-H 4-Channel Adapter***	4674536264A	AMI	3	1,750	12	21,000	1,050
Netfinity Fibre Channel RAID Controller Unit	35261RU		1	12,560	1	12,560	1,980
Netfinity Fibre Channel PCI Adapter	01K7297		1	1,675	1	1,675	0
Netfinity Fibre Channel 5M Short-Wave Cable	03K9306		1	127	1	127	0
9.1GB 10K Wide Ultra SCSI Drive	36L9748		1	469	1	469	0
Netfinity 4.2M Ultra2 SCSI Cable	03K9311		1	121	44	5,324	0
Ethernet 10/100 PCI Adapter	34L0201		1	58	1	58	0
IBM G42 14" (13.2" Viewable) Color Monitor*	654000N		1	169	1	169	215
20/40GB DLT SCSI Internal Tape Drive	01K1320		1	1,955	1	1,955	0
Netfinity Rack*	9306900		1	1,685	4	6,740	0
Side Panel Kit	94G6669		1	197	1	197	0
					Subtotal	$136,066	$13,540
Storage Hardware							
Netfinity EXP200 Rack Storage Enclosure	35301RU		1	2,510	42	105,420	83,160
9.1GB 10K Wide Ultra SCSI Drive	36L9748		1	469	400	187,600	0
18.2GB 10K Wide Ultra SCSI Drive	36L9749		1	837	12	10,044	0
					Subtotal	$303,064	$83,160
Server Software							
Microsoft SQL Server 7.0, Enterprise Edition, unlimited CALs		Microsoft	2	28,999	1	$28,999	$0
Microsoft Windows NT Server 4.0, Enterprise Edition, plus 25 CALs		Microsoft	2	3,999	1	3,999	0
Five-Year Maintenance for Software		Microsoft	2	10,475	1		10,475
					Subtotal	$32,998	$10,475
Client Hardware							
IBM Netfinity 5000 / 600MHz/512KB Pentium III*	86596RY		1	3,035	6	$18,210	$34,170
9.1GB 10K Wide Ultra SCSI Drive	36L9748		1	471	6	2,826	0
128MB DIMMs	01K7262		1	417	18	7,506	0
Netfinity 10/100 Ethernet PCI Adapter	34L1501		1	101	12	1,212	0
Intel Pro/100+ Dual-Port Ethernet Adapter**		Intel	1	69	12	828	0
IBM G42 14" (13.2" Viewable) Color Monitor*	654000N		1	169	6	1,014	1,290
					Subtotal	$31,596	$35,460
Client Software							
Microsoft Windows 2000 Server with COM+		Microsoft	2	809	6	4,854	0
Microsoft Visual C++ Professional 6.0		Microsoft	2	549	1	549	0
					Subtotal	$5,403	$0
User Connectivity							
8-Port 10Mbps Hub***	DEH2924	Generic	1	25	4,475	111,875	0
NETLUX 8-Port 100Mbps Hub***	NX-H8TX	NETLUX	4	95	3	285	0
					Subtotal	$112,160	$0
					Total	$621,287	$142,635

Notes: * The standard 3-year warranty and the extended warranty on IBM hardware is for 7x24, on-site same day coverage. ** Five-year warranty. *** 10% or minimum 2 spares are added in place of on-site service (products have a 5-year return-to-vendor-warranty)

Pricing: 1 - Software House International; 2 - Microsoft Corp.; 3 - American Megatrends.; 4 - NETLUX

Audited by Francois Raab of Info Sizing.

Five-Year Cost of Ownership:	$763,922
tpmC Rating:	40,251.15
$ / tpmC:	$18.98

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specification. If you find that stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.

Numerical Quantities Summary

MQTh, Computed Maximum Qualified Throughput:	40,251.15 tpmC
% throughput difference, reported and reproducibility runs:	1.35 %

Response Times (in seconds)	90%	Average	Maximum
New-Order	0.83	0.53	5.48
Payment	0.67	0.38	3.21
Order-Status	0.68	0.39	5.19
Delivery (Interactive)	0.13	0.12	1.04
Delivery (Deferred)	0.79	0.57	1.78
Stock-Level	2.72	2.29	6.78
Menu	0.13	0.12	1.49

Transaction Mix (in percent of total transactions)	Total Occurrences	Percent
New-Order	805,023	44.80
Payment	773,672	43.06
Order-Status	72,631	4.04
Delivery	73,118	4.07
Stock-Level	72,472	4.03

Emulation Delay (in seconds)		Response Time	Menu
New-Order		0.1	0.1
Payment		0.1	0.1
Order-Status		0.1	0.1
Delivery		0.1	0.1
Stock-Level		0.1	0.1

Keying/Think Times (in seconds)	Minimum	Average	Maximum
New-Order	18.01 / 0.00	18.03 / 12.01	18.08 / 120.32
Payment	3.01 / 0.00	3.03 / 12.01	3.08 / 120.32
Order Status	2.01 / 0.00	2.03 / 10.02	2.07 / 100.40
Delivery	2.01 / 0.00	2.03 / 5.04	2.08 / 50.31
Stock-Level	2.01 / 0.00	2.03 / 5.04	2.07 / 50.31

Test Duration	
Ramp-up time	47 minutes
Measurement interval	20 minutes
Number of transactions (all types) completed in measurement interval	1,796,916
Ramp-down time	11 minutes
Number of checkpoints in measurement interval	1
Checkpoint interval	20 minutes

Table of Contents

Preface

The TPC Benchmark™ C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark C Standard Specification Version 3.5, released October 25, 1999.

The TPC describes this benchmark in Clause 0.1 of the specification as follows:

TPC Benchmark C is an On Line Transaction Processing (OLTP) workload. It is a mixture of read-only and update-intensive transactions that simulate the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with environments, which are characterized by:

- The simultaneous execution of multiple transaction types that span a breadth of complexity
- On-line and deferred transaction execution modes
- Multiple on-line terminal sessions
- Moderate system and application execution time
- Significant disk input/output
- Transaction integrity (ACID properties)
- Non-uniform distribution of data access through primary and secondary keys
- Databases consisting of many tables with a wide variety of sizes, attributes and relationships
- Contention on data access and update

The performance metric reported by TPC-C is a “business throughput” measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C (tpmC). To be compliant with the TPC-C standard, all references to tpmC results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

Benchmark results are highly dependent upon workload, specific application requirements, and systems design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC-C should not be used as a substitute for a specific customer application benchmarking when critical capacity planning and/or product evaluation decisions are contemplated.

General Items

Benchmark Sponsor

A statement identifying the benchmark sponsor(s) and other participating companies must be provided.

This benchmark was sponsored by International Business Machines Corporation.

Application Code Disclosure and Definition Statements

The application program (as defined in Clause 2.1.7) must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input and output functions.

Appendix A contains all source code implemented in this benchmark.

Parameter Settings

Settings must be provided for all customer-tunable parameters and options that have been changed from the defaults found in actual products, including but not limited to:

- *Database tuning options*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and application configuration parameters.*
- *Compilation and linkage options and run-time optimizations used to create/install applications, OS, and/or databases.*

This requirement can be satisfied by providing a full list of all parameters and options.

Appendix C contains the tunable parameters for the database, the operating system, and the transaction monitor.

Configuration Diagrams

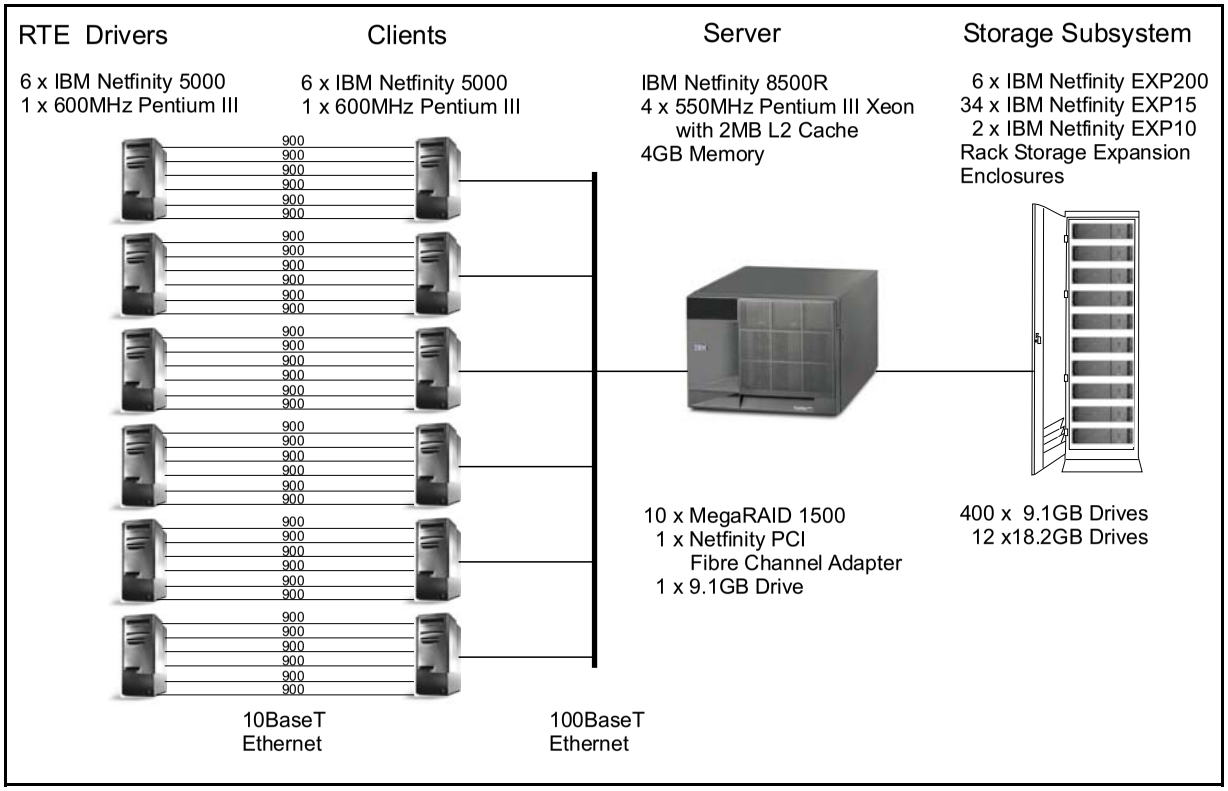
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

The configuration diagrams for the tested and priced systems are provided on the following pages.

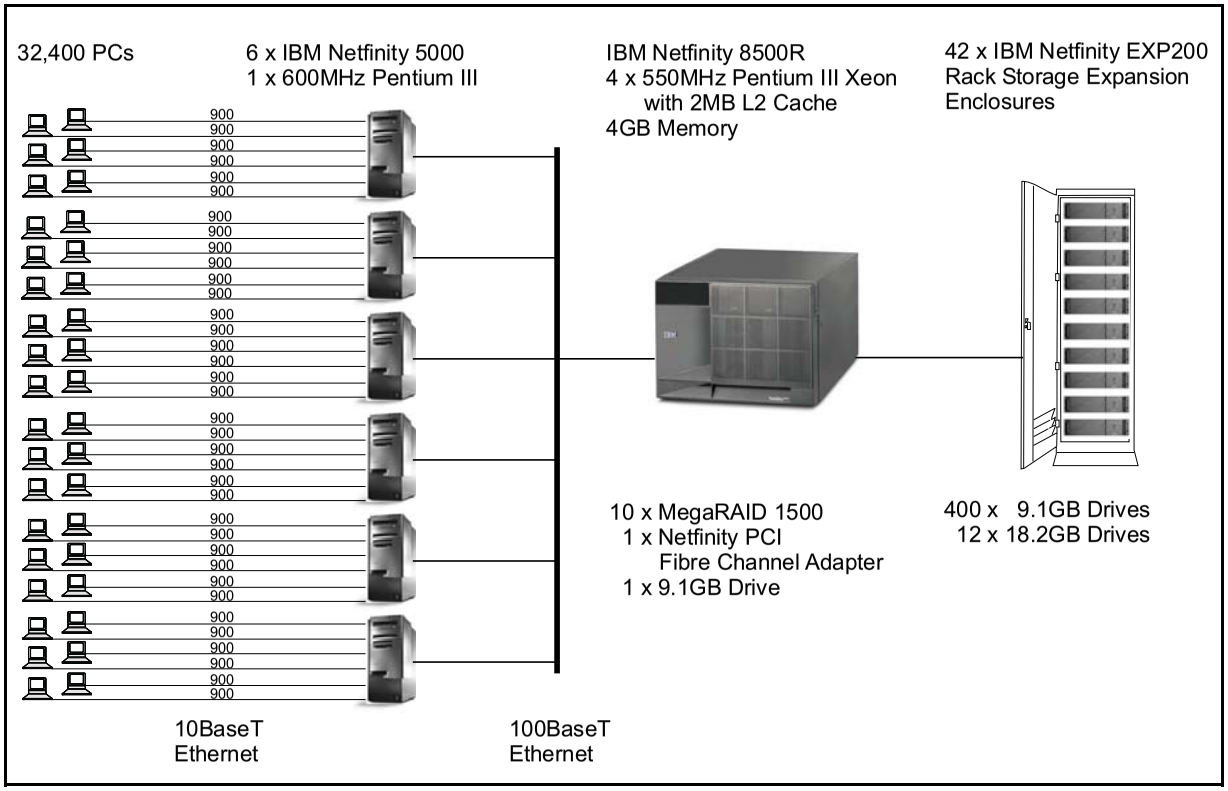
The Remote Terminal Emulator (RTE) used for these TPC Benchmark C tests is the Microsoft BenchCraft RTE. The components of the configuration being emulated by the RTE are the workstations and the Ethernet hubs. Appendix C contains a listing of the RTE scripts and inputs used in the benchmark testing.

The benchmarked configuration used IBM Netfinity 5000 systems as clients, which executed the terminal I/O and submitted transactions to COM+ servers, which are also running on the clients. These COM+ servers forwarded the transaction requests to the server, and returned the results to the RTE. Microsoft SQL Server Enterprise Edition 7.0 is the DBMS executing on the server

IBM Netfinity 8500R Measured Configuration



IBM Netfinity 8500R Priced Configuration



The priced and measured configurations were identical with two exceptions.

Drives and Storage Substitutions

The measured configuration used a combination of Netfinity EXP15 and EXP200 Rack Storage Expansion Enclosures. These two enclosures contain only passive components (i.e., power supplies and fans), which have no impact on performance. A combination of two different 9.1GB drives was used.

To verify equivalent performance of the drives and enclosures, tests were run and the data monitored as follows:

One AMI MegaRAID controller and forty 9.1GB drives (part number 36L9748) in four EXP200 enclosures were tested. Similarly, one AMI MegaRAID controller and forty 9.1GB drives (part number 01K8499) in four EXP15 enclosures were tested. The performance monitor data showed that the disk latency and disk queue depth for each configuration were equivalent.

Network Configuration

In the measured configuration, six physical 10Mbps Ethernet hubs were used to connect one RTE-client pair. For the other five RTE-client pairs, cross-over cables were used to emulate the connections to the Ethernet hubs. BenchCraft was used to generate separate transaction reports for each RTE-client pair. The response time data indicated no difference in performance between the use of hubs and cross-over cables. The data was reviewed by the auditor.

Clause 1: Logical Database Design Related Items

Table Definitions

Listings must be provided for all table definition statements and all other statements used to set up the database. (8.1.2.1)

Appendix B contains the code used to define and load the database tables.

Physical Organization of the Database

The physical organization of tables and indexes within the database must be disclosed. (8.1.2.2)

Physical space was allocated to Microsoft SQL Server on the server disks as detailed in Figure 4-2.

Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restriction in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows. (8.1.2.3)

All insert and delete functions were fully operational during the running of the benchmark. The space required for an additional 5 percent of the initial table cardinality was allocated to Microsoft SQL Server and priced as static space.

Horizontal or Vertical Partitioning

While there are few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark (see Clause 1.6), any such partitioning must be disclosed. (8.1.2.4)

Partitioning was not used in this benchmark.

Replication

Replication tables, if used, must be disclosed (see Clause 1.4.6). (8.1.2.5)

Replication was not used in this benchmark.

Table Attributes

Additional and/or duplicated attributes in any table must be disclosed, along with a statement on the impact on performance (see Clause 1.4.7). (8.1.2.6)

No additional attributes were used in this benchmark.

Clause 2: Transaction and Terminal Profiles Related Items

Random Number Generation

The method of verification for the random number generation must be disclosed.

The seeds and offsets for the random number generator were collected and verified to be different for each driver. The auditor selected samples of the generated numbers from the database. The samples were verified to have no discernible patterns.

Screen Layout

The actual layouts of the terminal input/out screens must be disclosed.

All screen layouts followed the TPC Benchmark C Standard Specification. .

Terminal Verification

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained. Although not specifically priced, the type and model of the terminals used must for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The auditor verified terminal features by direct experimentation. The benchmarked configuration uses Microsoft Internet Explorer 5.0 and HTML scripts as the terminal interface.

Intelligent Terminals

Any usage of presentation managers or intelligent terminals must be explained.

The terminals emulated in the priced configuration are IBM PC desktop computer systems. All processing of the input/output screens was handled by the IBM Netfinity 5000 clients. The screen input/output was managed via HTML strings that comply with the HTML Version 2.0 specification. A listing of the code used to implement the intelligent terminals is provided in Appendix A. All data manipulation was handled by the IBM Netfinity 8500R.

Transaction Profiles

The percentage of home and remote order-lines in the New-Order transactions must be disclosed. (8.1.3.5)
The percentage of New-Order transactions that were rolled back as a result of an unused item number must be disclosed. (8.1.3.6)
The number of items per orders entered by New-Order transactions must be disclosed. (8.1.3.7)
The percentage of home and remote Payment transactions must be disclosed. (8.1.3.8)
The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed. (8.1.3.9)
The percentage of Delivery transactions that were skipped as a result of an insuffic ient number of rows in the NEW-ORDER table must be disclosed. (8.1.3.10)
The mix (i.e., percentages) of transaction types seen by the SUT must be disclosed. (8.1.3.11)

Table 2-1. Transaction Statistics

New Order	Value (%)
Home warehouse order lines	99.00
Remote warehouse order lines	1.00
Rolled back transactions	1.00
Average number of items per order	10.01
Payment	
Home warehouse payment transactions	84.95
Remote warehouse payment transactions	15.05
Non-Primary Key Access	
Payment transactions using C_LAST	60.12
Order-Status transactions using C_LAST	60.31
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.80
Payment	43.06
Order-Status	4.04
Stock-Level	4.03
Delivery	4.07

Deferred Delivery Mechanism

The queuing mechanism used to defer execution of the Delivery transaction must be disclosed. (8.1.3.12)

The deferred delivery operation is queued by making an entry in an array within the application process (tpcc.dll) running on the client. Background threads within the application asynchronously process the queued delivery transactions.

The source code is listed in Appendix A.

Clause 3: Transaction and System Properties Related Items

The results of the ACID test must be disclosed, along with a description of how the ACID requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7. (8.1.4.1)

Atomicity Requirements

The system under test must guarantee that database transactions are atomic; the system will either perform all individual operations on the data, or will assure that no partially completed operations leave any effects on the data.

All ACID tests were conducted according to specification.

Completed Transactions

The following steps were performed to verify the Atomicity of completed transactions.

1. The balance was retrieved from the CUSTOMER table for a random Customer, District and Warehouse, giving BALANCE_1.
2. The Payment transaction was executed for the Customer, District and Warehouse used in step 1.
3. The balance was retrieved again for the Customer used in step 1 and step 2, giving BALANCE_2. It was verified that BALANCE_1 was greater than BALANCE_2 by AMT.

Aborted Transactions

The following steps were performed to verify the Atomicity of the aborted Payment transaction:

1. The Payment application code was changed to execute a rollback of the transaction instead of performing the commit.
2. Using the balance, BALANCE_2, from the CUSTOMER table retrieved for the completed transaction, the Payment transaction was executed for the Customer, District and Warehouse used in step 1 of section 3.1.1. The transaction rolled back due to the change in the application code from step 1.
3. The balance was retrieved again for the Customer used for step 2, giving BALANCE_3. It was verified that BALANCE_2 was equal to BALANCE_3.

Consistency Requirements

Consistency is the property of the application that requires any execution of a database transaction to take the database from one consistent state to another, assuming that the database is initially in a consistent state.

Consistency conditions one through four were tested using a shell script to issue queries to the database.The results of the queries demonstrated that the database was consistent for all four tests.

Isolation Requirements

Sufficient conditions must be enabled at either the system or the application level to ensure that the required isolation defined in Clause 3.4.1 is obtained.

Isolation tests one through seven were run using shell scripts to issue queries to the database. Each script included timestamps to demonstrate the concurrency of operations. The results of the queries were captured and placed in files. The auditor reviewed the results and verified that the isolation requirements had been met.

In addition, the phantom tests and the stock-level tests were run and verified.

Case A was followed for Isolation test seven.

Durability Requirements

The tested system must guarantee durability: the ability to preserve the effects of committed transactions and ensure database consistency after recovery from any one of the failures listed in Clause 3.5.3.

- *Permanent irrecoverable failure of any single durable medium containing TPC-C database tables or recovery log data (this test includes failure of all or part of memory)*
- *Instantaneous interruption (system crash/system hang) in processing that requires system reboot to recover*
- *Failure of all or part of memory (loss of contents)*

Loss of Data Test

The following steps were successfully performed to pass the Durability test of failure of a disk unit with database tables:

1. The contents of the database were backed up to several database dump devices during the initial database load. There were no dump devices on the disk array from which a drive was removed as part of this test.
2. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID for all rows in the district table giving SUM1.
3. A test was started with 100 users submitting transactions.
4. A disk containing a portion of each of the tables in the tpcc database was removed causing SQL Server to report errors accessing that device.
5. The run was aborted and SQL Server was restarted. Upon restart, the database tpcc reported numerous errors relating to the failed database device.
6. The transaction log was dumped to disk and the failed disk was replaced with a spare disk and was recovered.
7. The database was recovered and restored from the backup dump devices. Afterwards, the transaction log was applied to the database.
8. Step 2 was repeated to obtain the current count of the total number of orders giving SUM2.
9. It was verified that the sum of D_NEXT _O_ID after the database is recovered is greater than or equal to the sum of D_NEXT_O_ID before the run, plus all new order transactions completed during the run minus any rollback transactions.
10. Consistency Condition 3 was verified.

Combined Loss of Log and Loss of System Test (Instantaneous Interruption and Loss of Memory)

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID for all rows in the district table gving SUM1.
2. A test was started under full load with all users submitting transactions.
3. One disk from the log array was removed. Since the disk was RAID-1 mirrored, SQL Server continued to process transactions without interruption.
4. The test continued under full load with all users submitting transactions. A checkpoint was issued, and the system continued to run for another 5 minutes.
5. The server under test was powered off, which removed power from the system and the memory.
6. The server was powered on again.
7. SQL Server was started to initiate automatic recovery from its log.
8. Step 1 was repeated to obtain the current count of the total number of orders giving SUM2.
9. It was verified that the sum of D_NEXT_O_ID after the database is recovered is greater than or equal to the sum of D_NEXT_O_ID before the run, plus all new order transactions completed during the run minus any rollback transactions.

Clause 4: Scaling and Database Population Related Items

Cardinality of Tables

The cardinality (e.g., the number of rows) of each table, as it existed at the start of the benchmark run (see Clause 4.2), must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table wer e deleted (see Clause 4.2.2), the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed. (8.1.5.1)

The database was originally built with 3,254 warehouses but the audited run used only 3,240 warehouses.

Table 4-1. Initial Cardinality of Tables

Table Name	Rows
Warehouse	3,254
District	32,540
Customer	97,620,000
History	97,620,000
Orders	97,620,000
New Order	29,286,000
Order Line	976,202,542
Stock	325,400,000
Item	100,000
Inactive Warehouses	14

Distribution of Tables and Logs

The distribution of tables and logs across all media must be explicitly depicted for the tested and priced systems. (8.1.5.2)

Figure 4-2 depicts the database configuration of the tested system to meet the 8-hour steady state requirement.

Figure 4-2. Data Distribution for the Benchmarked Configuration

Controller	Drives	Partition	Size	Use
1	40 - 9.1GB	E: F:	19250MB 9650MB	Customer and Stock Misc.
2	40 - 9.1GB	G:, H:	19250MB, 9650MB 318180MB (NTFS)	Customer and Stock, Misc. backup2
3	40 - 9.1GB	I: J:	19250MB 9650MB	Customer and Stock Misc.
4	40 - 9.1GB	K: L:	19250MB 9650MB	Customer and Stock Misc.
5	40 - 9.1GB	M: N:	19250MB 9650MB	Customer and Stock Misc.
6	40 - 9.1GB	O: P:	19250MB 9650MB	Customer and Stock Misc.
7	40 - 9.1GB	Q: R:	19250MB 9650MB	Customer and Stock Misc.
8	40 - 9.1GB	S: T:	19250MB 9650MB	Customer and Stock Misc.
9	40 - 9.1GB	U:, V: Z:	19250MB, 9650MB 318180MB (NTFS)	Customer and Stock, Misc. backup1
10	40 - 9.1GB	W: X:	19250MB 9650MB	Customer and Stock Misc.
11	12 - 18.2GB	Y:	61000MB (RAID-1)	Log File
12	1 - 9.1GB	C:	4095MB (NTFS)	OS

Database Model Implemented

A statement must be provided that describes:

1. *The database model implemented by the DBMS used (e.g., relational, network, hierarchical)*
2. *The database interface (e.g., embedded, call level) and access language (e.g., SQL, DL/1, COBOL, read/write) used to implement the TPC-C transactions. If more than one interface/access language is used to implement TPC-C, each interface/access language must be described and a list of which interface/access language is used with which transaction type must be disclosed.*

Microsoft SQL Server 7.0, Enterprise Edition, is a relational database. The interface used was Microsoft SQL Server stored procedures accessed with Remote Procedure Calls embedded in C code using the Microsoft DBLIB interface.

Partitions/Replications Mapping

The mapping of database partitions/replications must be explicitly described.

The database was neither partitioned nor replicated.

180-Day Space Requirement

Details of the 180-day space computations, along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed (see Clause 4.2.3). (8.1.5.5)

See Appendix E for details about how the 180-day space requirements were calculated.

Clause 5: Performance Metrics and Response Time Related Items

Measured tpmC

Measured tpmC must be reported. (8.1.6.1)

Measured tpmC: 40,251.15 tpmC

Price per tpmC: $18.98 per tpmC

Response Times

Ninetieth percentile, maximum and average response times must be reported for all transaction types as well as for the Menu response time. (8.1.6.2)

The TPC-C requirements for the average response time and the 90th percentile were met. Table 5-1 provides the response times for each of the transaction types and the menu for the measured system.

Table 5-1. Response Times in Seconds

Transaction Type	Average	Maximum	90th %
New-Order	0.53	5.48	0.83
Payment	0.38	3.21	0.67
Order-Status	0.39	5.19	0.68
Delivery (interactive)	0.12	1.04	0.13
Delivery (deferred)	0.57	1.78	0.79
Stock-Level	2.29	6.78	2.72
Menu	0.12	1.49	0.13

Keying/Think Times

The minimum, the average, and the maximum keying and think times must be reported for each transaction type. (8.1.6.3)

Table 5-2 lists the keying/think times for the measured system.

Table 5-2. Keying/Think Times

Transaction Type	Minimum	Average	Maximum
New-Order	18.01 / 0.00	18.03 / 12.01	18.08 / 120.32
Payment	3.01 / 0.00	3.03 / 12.01	3.08 / 120.32
Order-Status	2.01 / 0.00	2.03 / 10.02	2.07 / 100.40
Delivery	2.01 / 0.00	2.03 / 5.04	2.08 / 50.31
Stock-Level	2.01 / 0.00	2.03 / 5.04	2.07 / 50.31

Response Time Frequency Distribution Curves

Response time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type. (8.1.6.4)

Figure 5-1. New-Order Transaction - Response Time Frequency Distribution

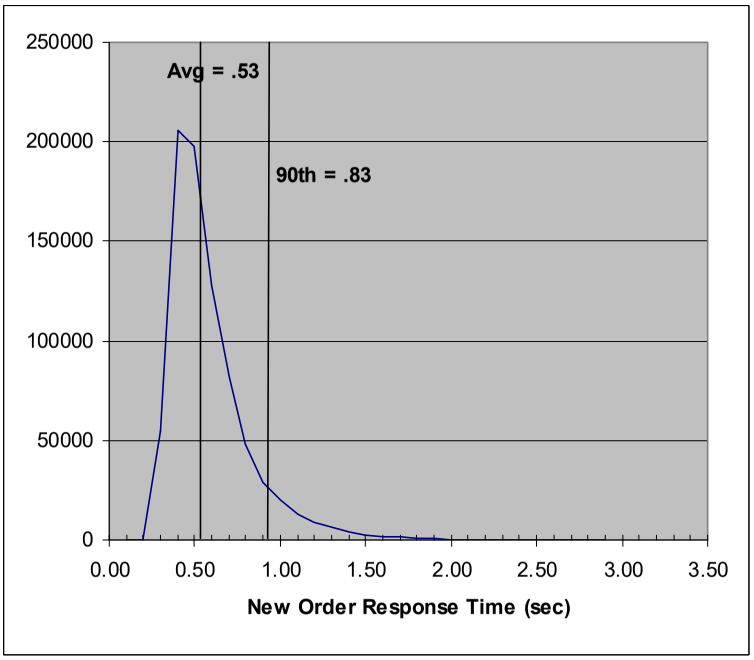


Figure 5-2. Payment Transaction - Response Time Frequency Distribution

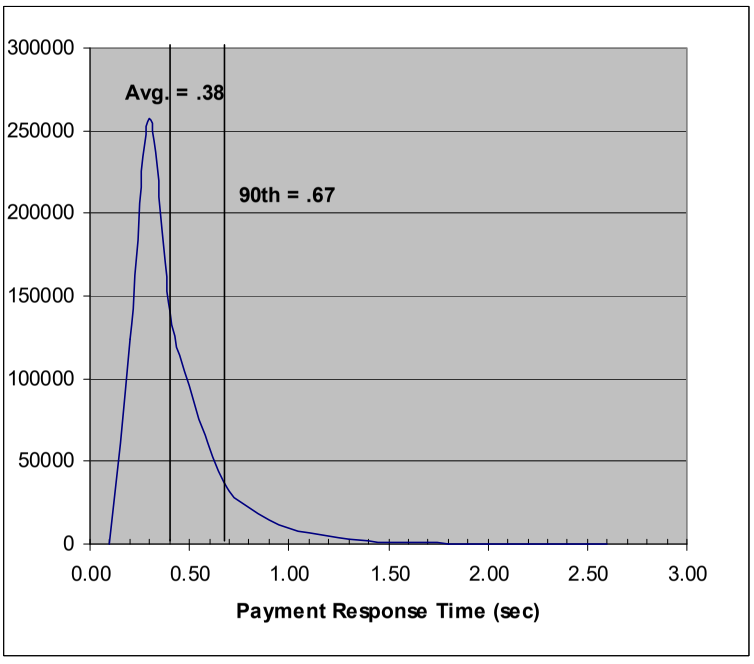


Figure 5-3. Order-Status Transaction - Response Time Frequency Distribution

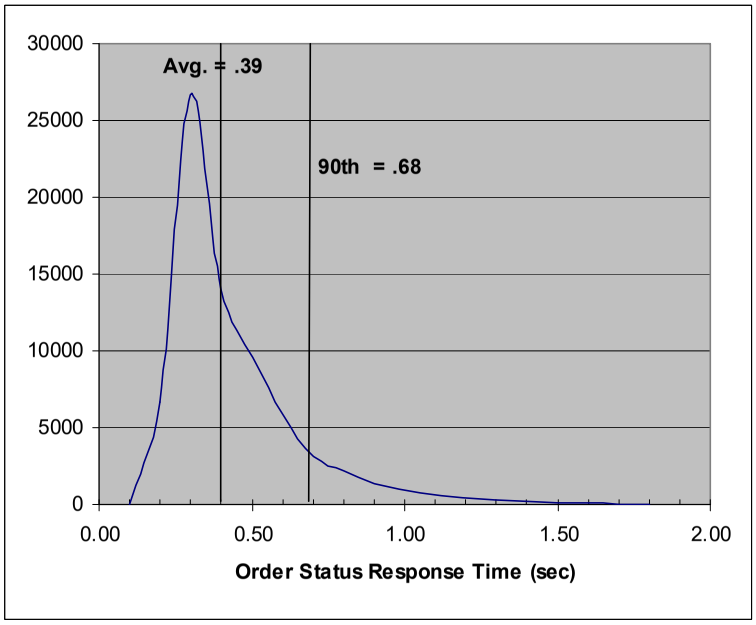


Figure 5-4. Delivery Transaction - Response Time Frequency Distribution

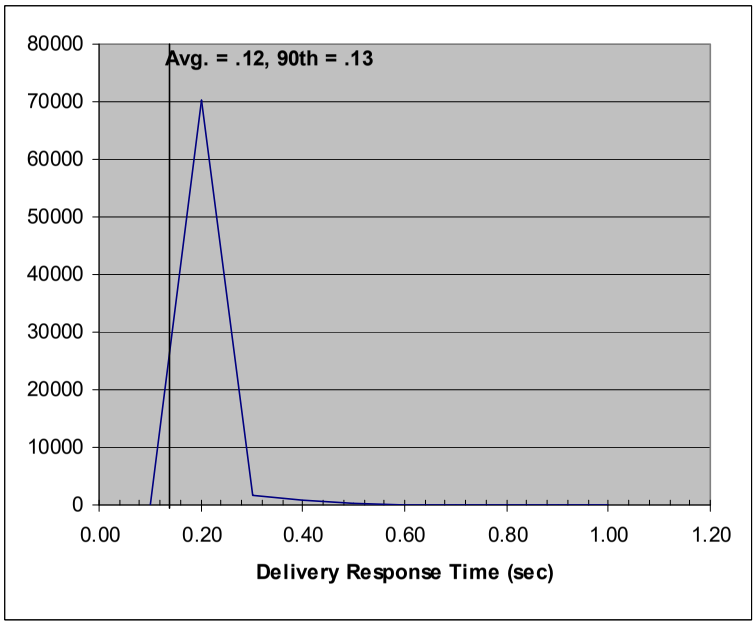
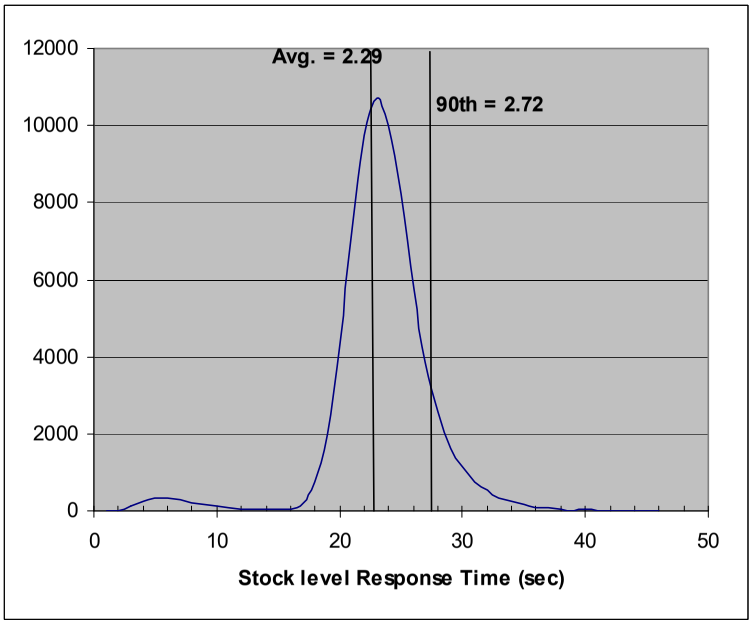


Figure 5-5. Stock-Level Transaction - Response Time Frequency Distribution



Performance Curve for Response Time vs. Throughput

The performance curve for response time vs. throughput (see Clause 5.6.2) must be reported for the New-Order transaction. (8.1.6.5)

Figure 5-6. New-Order Response Time vs. Throughput

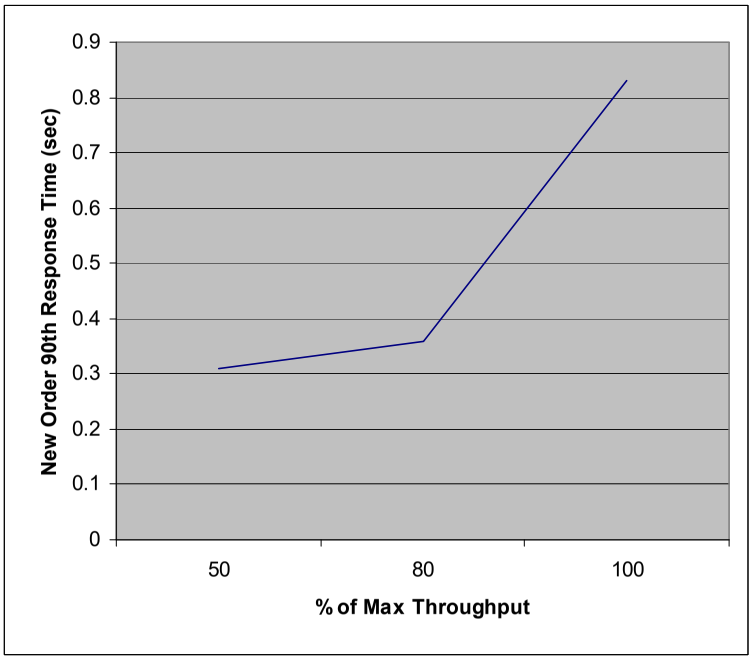
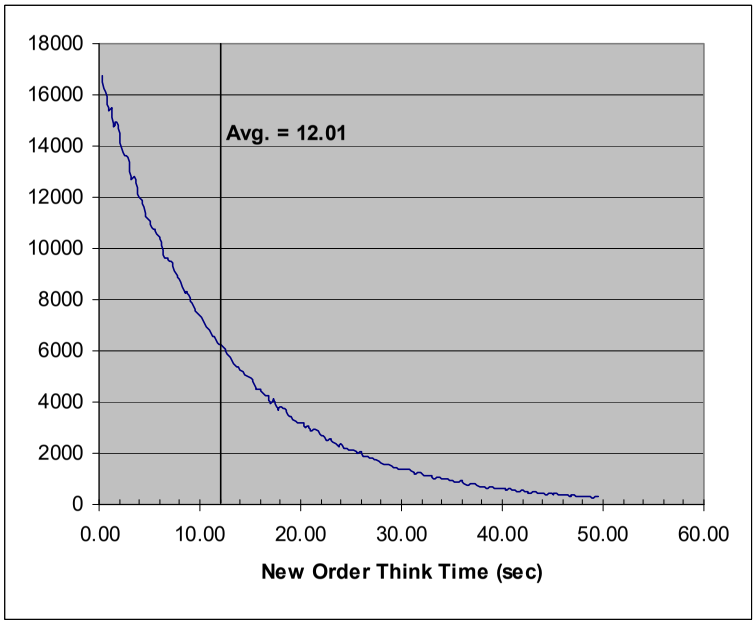


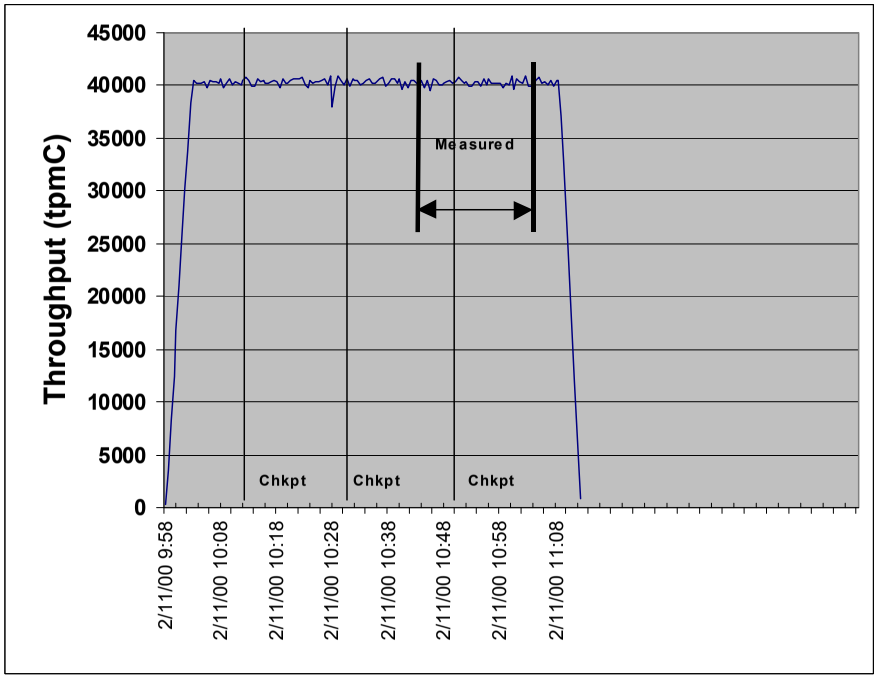
Figure 5-7. New-Order Think Time Distribution



Throughput vs. Elapsed Time

A graph of throughput vs. elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction.

Figure 5-8. New-Order Throughput vs. Elapsed Time



Steady State Methodology

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval (see Clause 5.5) must be described. (8.1.6.9)

Figure 5-8 shows that the system was in steady state at the beginning of the measurement interval.

Work Performed during Steady State

A description of how the work normally performed during a sustained test (e.g., checkpointing, writing redo/undo log records) actually occurred during the measurement interval must be reported. (8.1.6.10)

Transaction Flow

The RTE generated the required input data to choose a transaction from the menu. This data was time-stamped. The response for the requested transaction was verified and time-stamped in the RTE log files.

The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the input screen. The transmission was time-stamped. The return of the screen with the required response data was time-stamped. The difference between these two time-stamps was the response time for that transaction and was logged in the RTE log.

The RTE then waited the required think time interval before repeating the process starting at selecting another transaction from the menu.

The RTE transmissions were sent to application processes running on the client machines through Ethernet LANs. These client application processes handled all screen I/O as well as all requests to the database on the server. The applications communicated with the database server over another Ethernet LAN using Microsoft SQL Server DBLIB library and RPC calls.

Checkpoints

Checkpoints were executed on the server during the ramp-up phase and at 20-minute intervals. Each measured run contained one checkpoint. SQL Server was started with trace flag 3502, which caused it to log the occurrence of the checkpoint. This information was used to verify that the checkpoints occurred at the appropriate times during the test run.

During a checkpoint, SQL Server flushes all dirty pages from its cache to disk. It places a record in the database transaction log indicating that the checkpoint has completed and that all transactions, which were committed prior to the checkpoint have been written to disk.

Reproducibility Methodology

A description of the method used to determine the reproducibility of the measurement results must be reported. (8.1.6.11)

A repeatability measurement was taken on the IBM Netfinity 8500R for the same length of time as the measured run. The repeatability measurement was 39,711.05 tpmC.

Measurement Interval

A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included. (8.1.6.12)

The measurement interval was 20 minutes.

Transaction Mix

The method of regulation of the transaction mix (e.g., card decks or weighted random distribution) must be described. If weighted distribution is used and the RTE adjusts the weights associated with each transaction type, the maximum adjustments to the weight from the initial value must be disclosed. (8.1.6.13)

See Table 5-3.

The RTE was given a weighted random distribution, which was not adjusted during the run.

Percentage of Total Mix

The percentage of the total mix for each transaction type must be disclosed.

See Table 5-3.

Table 5-3. Transaction Statistics and Transaction Mix

New Order	Value (%)
Home warehouse order lines	99.00
Remote warehouse order lines	1.00
Rolled back transactions	1.00
Average number of items per order	10.01
Payment	
Home warehouse payment transactions	84.95
Remote warehouse payment transactions	15.05
Non-Primary Key Access	
Payment transactions using C_LAST	60.12
Order-Status transactions using C_LAST	60.31
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.80
Payment	43.06
Order-Status	4.04
Stock-Level	4.03
Delivery	4.07

Number of Checkpoints

The number of checkpoints in the Measurement Interval, the time in seconds from the start of the Measurement Interval to the first checkpoint, and the Checkpoint Interval must be disclosed.

Checkpoints were performed during the ramp-up period and during each measured run interval. The measured interval checkpoint started 5 minutes and 16 seconds after the start of the measurement interval. The checkpoint in the measured interval lasted 4 minutes and 19 seconds.

The checkpoints were verified to be clear of the protected zones around the beginning and end of the measurement intervals. The checkpoint interval was 20 minutes.

Clause 6: SUT, Driver and Communication Definition Related Items

Description of RTE

The RTE input parameters, code fragments, functions, etc., used to generate each transaction input field must be disclosed. (8.1.7.1)

The RTE used was Microsoft BenchCraft RTE. Benchcraft is a proprietary tool provided by Microsoft and is not commercially available. The RTE input is listed in Appendix A.

Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to that of the priced system. The results of the test described in Clause 6.6.3.4 must be disclosed. (8.1.7.2)

No components were emulated.

Benchmarked and Targeted System Configuration Diagrams

A complete functional diagram of both the benchmarked configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all software and hardware functionality being performed on the Driver System, and its interface to the SUT must be disclosed (see Clause 6.6.3.6).

The driver RTE generated the transaction input data and transmitted it to the client in HTML format. The driver RTE received the output from the System under Test, time-stamped it, and forwarded it to the Master RTE for post-test processing. No other functionality was included on the driver RTE.

Detailed diagrams of the benchmarked and priced configurations are provided in the section called "General Items" at the beginning of this document.

Network Configuration

The network configurations of both the tested services and the proposed (target) services which are being represented and a thorough explanation of exactly which parts of the proposed configuration are being replaced with the Driver System must be disclosed (see Clause 6.6.4). (8.1.7.4)

In the measured configuration, six physical 10Mbps Ethernet hubs were used to connect one RTE-client pair. For the other five RTE-client pairs, cross-over cables were used to emulate the connections to the Ethernet hubs. BenchCraft was used to generate separate transaction reports for each RTE-client pair. The response time data indicated no difference in performance between the use of hubs and cross-over cables. The data was reviewed by the auditor.

Network Bandwidth

The bandwidth of the network(s) used in the tested/priced configuration must be disclosed. (8.1.7.5)

The Ethernet used in the LAN complies with the IEEE.802.3 standard. The LANs that connected the driver RTEs to the clients had a bandwidth of 10Mbps. The LAN that connected the clients to the server had a bandwidth of 100Mbps.

Operator Intervention

If the configuration requires operator intervention (see Clause 6.6.6), the mechanism and the frequency of this intervention must be disclosed. (8.1.7.6)

The configuration did not require any operator intervention to sustain the reported throughput during the eight-hour period.

Clause 7: Pricing Related Items

Hardware and Software Components

A detailed list of the hardware and software used in the priced system must be reported. Each separately orderable item must have a vendor part number, description and release/revision level, and either general availability status or committed delivery date. If package-pricing is used, vendor part number of the package and a description uniquely identifying each of the components of the package must be disclosed.

Pricing source(s) and effective date(s) must also be reported. (8.1.8.1)

The total 5-year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed. (8.1.8.2)

A detailed list of all hardware and software, including the 5-year price, is provided in the Executive Summary at the front of this report. All third-party quotations are included in Appendix F at the end of this document.

Availability Date

The committed delivery date for general availability (availability date) of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the reported availability for the priced system must be the date at which all components are committed to be available. (8.1.8.3)

All hardware and software used in this benchmark are currently available.

Measured tpmC

A statement of the measured tpmC, as well as the respective calculations for the 5-year pricing, price/performance (price/tpmC) and the availability date must be included. (8.1.8.4)

- Maximum Qualified Throughput: 40,251.15 tpmC
- Price per tpmC: $18.98 per tpmC
- Five-year cost of ownership: $763,922

Country-Specific Pricing

Additional Clause 7 related items may be included in the Full Disclosure Report for each country-specific priced configuration. Country-specific pricing is subject to Clause 7.1.7.

The configuration is priced for the United States of America.

Usage Pricing

For any usage pricing, the sponsor must disclose:

- *Usage level at which the component was priced.*
- *A statement of the company policy allowing such pricing. (8.1.8.6)*

The component pricing based on usage is shown below:

- 6 Microsoft Windows 2000 Server 4.0 (one server includes 25 CALs)
- 1 Microsoft Windows NT Server 4.0, Enterprise Edition, including 25 CALs
- 1 Microsoft SQL Server 7.0, Enterprise Edition (one server plus 25 CALs)

- 5-year support for hardware components (except for components for which 10 percent spares are provided)

System Pricing

System pricing should include subtotals for the following components: Server Hardware, Server Software, Client Hardware, Client Software, and Network Components used for terminal connection (see Clause 7.2.2.3). (8.1.8.7) System pricing must include line item indication where non-sponsoring companies' brands are used. System pricing must also include line item indication of third-party pricing. (8.1.8.8)

A detailed list of all hardware and software, including the 5-year price, is provided in the Executive Summary at the front of this report. All third-party quotations are included in Appendix F at the end of this document.

Clause 9: Audit Related Items

Auditor

The auditor's name, address, phone number, and a copy of the auditor's attestation letter indicati ng compliance must be included in the Full Disclosure Report. (8.1.9.1)

This implementation of the TPC-C benchmark was audited by Francois Raab of Info Sizing. The auditor's attestation letter is provided in this section.

Availability of the Full Disclosure Report

The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to the charges for similar documents by the test sponsor. The report must be made available when results are made public. In order to use the phrase "TPC Benchmark™C," the Full Disclosure Report must have been submitted to the TPC Administrator as well as written permission obtained to distribute same. (8.2)

Requests for the TPC Benchmark C Full Disclosure Report should be sent to:

Transaction Processing Performance Council
c/o Shanley Public Relations
650 N. Winchester Blvd., Suite 1
San Jose, CA 95128

or

IBM Corporation
Mail Drop 23U/060/D133
3039 Cornwallis Road
Research Triangle Park, NC 27709

INFOSIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Benchmark Sponsor: William D. Hall
Mgr., Netfinity Performance
IBM Personal Systems Group
3039 Cornwallis Road
Research Triangle Park, NC 27709

February 24, 2000

I verified the TPC Benchmark™ C performance of the following Client Server configuration:

Platform:	**IBM Netfinity 8500R c/s**
Operating system:	**Microsoft SQL Server 7.0 Enterprise Edition**
Database Manager:	**Microsoft Windows NT 4.0 Enterprise Edition**
Transaction Manager:	**Microsoft Com+**

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	**tpmC**
Server: IBM Netfinity 8500R				
8 x Pentium III Xeon (550 Mhz)	4.0 GB (2MB L2 cache per processor)	401 x 9.1 GB 12 x 18.2 GB	.83 Seconds	**40,251.15**
Six (6) Clients: IBM Netfinity 5000 (Specification for each)				
1 x Pentium III (600 Mhz)	512 MB	1 x 9.1 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC's requirements for the benchmark. The following verification items were given special attention:

- The database records were the proper size
- The database was properly scaled and populated
- The required ACID properties were met

1373 North Franklin Street Colorado Springs, CO 80903-2527 Office: 719/473-7555 Fax: 719/47

- The transactions were correctly implemented
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- All 90% response times were under the specified maximums
- At least 90% of all delivery transactions met the 80 Second completion time limit
- The reported measurement interval was 20 minutes (1200 seconds)
- The reported measurement interval was representative of steady state conditions
- One checkpoint was taken during the reported measurement interval
- The repeatability of the measured performance was verified
- The 180 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

The measured system included (360) Cheetah II (9.1 GB disks) that were substituted by (360) Cheetah III (9.1 GB disks) in the priced configuration. Based on the specifications of these disks and on additional performance data collected on these disks, it is my opinion that this substitution does not have a material effect on the reported performance.

The measured system was connected to the RTE using crossover cables instead of the network hubs included in the priced configuration. Based on additional performance data collected on these two types of interconnect, it is my opinion that this substitution does not have a material effect on the reported performance.

Respectfully Yours,

François Raab
President

1373 North Franklin Street Colorado Springs, CO 80903-2527 Office: 719/473-7555 Fax: 719/47

Appendix A: Source Code

db_dblib_dll.dsp

```
# Microsoft Developer Studio Project File - Name="db_dblib_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=db_dblib_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "db_dblib_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "db_dblib_dll.mak" CFG="db_dblib_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "db_dblib_dll - Win32 Release" (based on\
 "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_dblib_dll - Win32 Debug" (based on\
 "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_dblib_dll - Win32 IceCAP" (based on\
 "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "db_dblib_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib /nologo /subsystem:windows /dll /machine:I386 /out:".\bin/tpcc_dblib.dll"

!ELSEIF  "$(CFG)" == "db_dblib_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib /nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_dblib.dll"/pdbtype:sept

!ELSEIF  "$(CFG)" == "db_dblib_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_dblib"
# PROP BASE Intermediate_Dir "db_dblib"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /Gh /c
```

```
# ADD CPP /nologo /MD /W3 /Gm /GX /Zi /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D "ICECAP" /YX /FD
/Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib /nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_dblib.dll"/pdbtype:sept
# ADD LINK32 icap.lib ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib /nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_dblib.dll"/pdbtype:sept

!ENDIF

# Begin Target

# Name "db_dblib_dll - Win32 Release"
# Name "db_dblib_dll - Win32 Debug"
# Name "db_dblib_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp"
# Begin Source File

SOURCE=.\src\tpcc_dblib.cpp
# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_dblib.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project
```

dlldata.c

```
/*********************************************************
  DllData file -- generated by MIDL compiler

      DO NOT ALTER THIS FILE

  This file is regenerated by MIDL on every IDL file compile.

  To completely reconstruct this file, delete it and rerun MIDL
  on all the IDL files in this DLL, specifying this file for the
  /dlldata command line option

*********************************************************/


#include <rpcproxy.h>

#ifdef __cplusplus
extern "C"   {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )


PROXYFILE_LIST_START
/* Start of list */
  REFERENCE_PROXY_FILE( tpcc_com_ps ),
/* End of list */
PROXYFILE_LIST_END


DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

#ifdef __cplusplus
}  /*extern "C" */
#endif

/* end of generated dlldata file */
```

error.h

```
/*              FILE:                           ERROR.H
 *                                                      Microsoft TPC-C Kit Ver. 4.20.000
 *                                                      Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                                      Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 *              PURPOSE:  Header file for error exception classes.
 *
 *  Change history:
 *                              4.20.000 - updated rev number to match kit
 */
```

```
#pragma once

#ifndef _INC_STRING
	#include <string.h>
#endif

const int m_szMsg_size = 512;
const int m_szApp_size = 64;
const int m_szLoc_size = 64;

//error message structure used in ErrorText routines
typedef struct _SERRORMSG
{
	int		iError;		//error id of message
	char	szMsg[256];	//message to sent to browser
} SERRORMSG;

#define ERR_FATAL_LEVEL		1
#define ERR_WARNING_LEVEL	2
#define ERR_INFORMATION_LEVEL	3


#define ERR_TYPE_LOGIC		-1
	//logic error in program; internal error
#define ERR_SUCCESS		0
	//success (a non-error error)
#define ERR_BAD_ITEM_ID		1
	//expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST	2
//expected delivery post failed
#define	ERR_TYPE_WEBDLL		3
	//tpcc web generated error
#define	ERR_TYPE_SQL		4
	//sql server generated error
#define	ERR_TYPE_DBLIB		5
	//dblib generated error
#define	ERR_TYPE_ODBC		6
	//odbc generated error
#define ERR_TYPE_SOCKET		7
	//error on communication socket client rte only
#define ERR_TYPE_DEADLOCK	8
//dblib and odbc only deadlock condition
#define ERR_TYPE_COM		9
	//error from COM call
#define ERR_TYPE_TUXEDO		10
	//tuxedo error
#define ERR_TYPE_OS		11
	//operating system error
#define ERR_TYPE_MEMORY		12
	//memory allocation error
#define ERR_TYPE_TPCC_ODBC	13
//error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB	14
//error from tpcc dblib txn module
#define ERR_TYPE_DELISRV	15
//delivery server error
#define ERR_TYPE_TXNLOG		16
	//txn log error
#define ERR_TYPE_BCCONN		17
	//Benchcraft connection class
#define ERR_TYPE_TPCC_CONN	18
//Benchcraft connection class
#define ERR_TYPE_ENCINA		19
	//Encina error
#define ERR_TYPE_COMPONENT	20
//error from COM component

class CBaseErr
{
public:
	char	*m_szApp;
	char	*m_szMsg;
	char	*m_szLoc;  // code location where the error occurred
	int		m_idMsg;

	CBaseErr(void)
	{
		m_idMsg		= 0;
		m_szMsg		= new char[m_szMsg_size];
		m_szApp		= new char[m_szApp_size];
		m_szLoc		= NULL;

		m_szMsg[0] = 0;
		m_szApp[0] = 0;

		GetModuleFileName(GetModuleHandle(NULL),m_szApp, m_szApp_size);
	}

	~CBaseErr(void)
	{
		if (m_szMsg)
			delete [] m_szMsg;
		if (m_szApp)
			delete [] m_szApp;
		if (m_szLoc)
			delete [] m_szLoc;
	};

	CBaseErr(int idMsg)
	{
		m_idMsg		= idMsg;
		m_szApp		= new char[m_szApp_size];
		m_szMsg		= new char[m_szMsg_size];
		m_szLoc		= NULL;

		GetModuleFileName(GetModuleHandle(NULL),m_szApp, m_szApp_size);
		LoadString(GetModuleHandle(NULL),idMsg, m_szMsg, m_szMsg_size);
	}

	CBaseErr(LPCTSTR szMsg)
	{
		m_idMsg		= 0;
```

```
                m_szApp                 = new char[m_szApp_size];
                m_szMsg                 = new char[m_szMsg_size];
                m_szLoc                 = NULL;

                GetModuleFileName(GetModuleHandle(NULL),m_szApp, m_szApp_size);
                strcpy(m_szMsg, szMsg);
        }

        void SetError(char *szMsg, LPCTSTR szLocation)
        {
                if (szMsg != NULL)
                        strcpy(m_szMsg, szMsg);
                else
                        m_szMsg[0] = 0;

                if (szLocation != NULL)
                {
                        delete [] m_szLoc;
                        m_szLoc = new char[strlen(szLocation)+1];
                        strcpy(m_szLoc, szLocation);
                }
                else
                {
                        delete [] m_szLoc;
                        m_szLoc = NULL;
                }
        }

        virtual void Draw(HWND hwnd, LPCTSTR szStr = NULL)
        {
                int             j;
                char    szTmp[512];

                if (szStr)
                        j = wsprintf(szTmp, "%s\n",szStr);
                if (m_szLoc)
                        j += wsprintf(szTmp+j, "Location=%s\n",m_szLoc);
                if (m_szMsg)
                        j += wsprintf(szTmp+j, "%s\n", m_szMsg);

                ::MessageBox(hwnd, szTmp, m_szApp, MB_OK);
        }

        char *GetApp(void) { return m_szApp; }
        char *GetMsg(void) { return m_szMsg; }
        char *GetLocation(void) { return m_szLoc; }

        virtual int ErrorType() = 0;            // a value which distinguishes the kind of error that occurred
        virtual int ErrorNum() = 0;                     // an error value specific to the error type
        virtual char *ErrorText() = 0;          // a string (i.e., human readable) representation of the error

};

class CSocketErr : public CBaseErr
{
public:
        enum Action
        {
                eNone,
                eSend,
                eSocket,
                eConnect
        };

        CSocketErr(Action eAction, LPCTSTR szLocation);
        CSocketErr(int iError) { m_errId = iError; };
        int                     m_errId;
        Action  m_eAction;

        int ErrorType() { return ERR_TYPE_SOCKET;};
        int ErrorNum() { return m_errId;};
        char *ErrorText(void);
};

class CSystemErr : public CBaseErr
{
public:
        enum Action
        {
                eNone,
                eTransactNamedPipe,
                eWaitNamedPipe,
                eSetNamedPipeHandleState,
                eCreateFile,
                eCreateProcess,
                eCallNamedPipe,
                eCreateEvent,
                eCreateThread,
                eVirtualAlloc,
                eReadFile,
                eWriteFile,
                eMapViewOfFile,
                eCreateFileMapping,
                eInitializeSecurityDescriptor,
                eSetSecurityDescriptorDacl,
                eCreateNamedPipe,
                eConnectNamedPipe,
        };

        CSystemErr(Action eAction, LPCTSTR szLocation);

        void Draw(HWND hwnd, LPCTSTR szStr = NULL);

        int                     m_errId;
        Action  m_eAction;

        int ErrorType() { return ERR_TYPE_OS;}
        int ErrorNum() { return m_errId;}
        char *ErrorText() { return "";}                 // TODO: need to code error text
};

class CMemoryErr : public CBaseErr
```

```
{
public:
        CMemoryErr(void);

        int ErrorType() { return ERR_TYPE_MEMORY;}
        int ErrorNum() { return 0;}
        char *ErrorText() { return "";}                 // TODO: need to code error text
};
```

install.c

```
/*      FILE:                   INSTALL.C
 *                                      Microsoft TPC-C Kit Ver. 4.20.000
 *                                      Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                                      not audited
 *
 *      PURPOSE:  Automated installation application forTPC-C Web Kit
 *      Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                      4.20.000 - added COM installation steps
 */

#include <windows.h>
#include <direct.h>
#include <io.h>
#include <stdlib.h>
#include <stdio.h>
#include <commctrl.h>
#include "..\..\common\src\ReadRegistry.h"

#include "resource.h"

#define WM_INITTEXT                     WM_USER+100

HICON           hIcon;
HINSTANCE       hInst;

DWORD                   versionExeMS;
DWORD                   versionExeLS;
DWORD                   versionExeMM;
DWORD                   versionDllMS;
DWORD                   versionDllLS;

// TPC-C registry settings
TPCCREGISTRYDATA Reg;

static  int             iPoolThreadLimit;
static  int             iThreadTimeout;
static  int             iListenBackLog;
static  int             iAcceptExOutstanding;

static  int             iMaxPhysicalMemory;             //max physical memory in MB
static  char    szLastFileName[64];             // last file we worked on (for error reporting)


BOOL    CALLBACK        LicenseDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam);
BOOL    CALLBACK        UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam);
BOOL    CALLBACK        MainDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
BOOL    CALLBACK        CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
static  void            ProcessOK(HWND hwnd, char *szDllPath);
static  void            ReadRegistrySettings(void);
static  void            WriteRegistrySettings(char *szDllPath);
static  BOOL            RegisterDLL(char *szFileName);
static  int                     CopyFiles(HWND hDlg, char *szDllPath);
static  BOOL            GetInstallPath(char *szDllPath);
static  void            GetVersionInfo(char *szDLLPath, char *szExePath);
static  BOOL            CheckWWWebService(void);
static  BOOL            StartWWWebService(void);
static  BOOL            StopWWWebService(void);
static  void            UpdateDialog(HWND hDlg);

BOOL install_com(char *szDllPath);

#include "..\..\common\src\ReadRegistry.cpp"

int WINAPI WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow )
{
        int iRc;

        hInst = hInstance;

        InitCommonControls();

        hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_ICON1));

        iRc = DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG4), GetDesktopWindow(),
LicenseDlgProc);
        if ( iRc )
        {
                iRc = DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG1), GetDesktopWindow(),
MainDlgProc);
                if ( iRc )
                {
                        DialogBoxParam(hInstance, MAKEINTRESOURCE(IDD_DIALOG2),
GetDesktopWindow(), UpdatedDlgProc, (LPARAM)iRc);
                }
        }

        DestroyIcon(hIcon);
        return 0;
}

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
```

```
{
	HGLOBAL			hRes;
	HRSRC			hResInfo;
	BYTE			*pSrc, *pDst;
	DWORD			dwSize;
	static HFONT		hFont;

	switch(uMsg)
	{
		case WM_INITDIALOG:
			hFont = CreateFont(-12, 0, 0, 0, 400, 0, 0, 0, 0, 0, 0, 0, 0,  "Arial");
			SendMessage( GetDlgItem(hwnd, IDR_LICENSE1), WM_SETFONT,
(WPARAM)hFont, MAKELPARAM(0, 0) );
			PostMessage(hwnd, WM_INITTEXT, (WPARAM)0, (LPARAM)0);
			return TRUE;
		case WM_INITTEXT:
			hResInfo = FindResource(hInst, MAKEINTRESOURCE(IDR_LICENSE1),
"LICENSE");
			dwSize = SizeofResource(hInst, hResInfo);
			hRes = LoadResource(hInst, hResInfo );
			pSrc = (BYTE *)LockResource(hRes);
			pDst = (unsigned char *)malloc(dwSize+1);
			if ( pDst )
			{
				memcpy(pDst, pSrc, dwSize);
				pDst[dwSize] = 0;
				SetDlgItemText(hwnd, IDC_LICENSE, (const char *)pDst);
				free(pDst);
			}
			else
				SetDlgItemText(hwnd, IDC_LICENSE, (const char *)pSrc);
			return TRUE;
		case WM_DESTROY:
			DeleteObject(hFont);
			return TRUE;
		case WM_COMMAND:
			if ( wParam == IDOK )
				EndDialog(hwnd, TRUE);
			if ( wParam == IDCANCEL )
				EndDialog(hwnd, FALSE);
		default:
			break;
	}
	return FALSE;
}

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
	switch(uMsg)
	{
		case WM_INITDIALOG:
			switch(lParam)
			{
				case 1:
				case 2:
					SetDlgItemText(hwnd, IDC_RESULTS, "TPC-C Web
Client Installed");
					break;
			}
			return TRUE;
		case WM_COMMAND:
			if ( wParam == IDOK )
				EndDialog(hwnd, TRUE);
			break;
		default:
			break;
	}
	return FALSE;
}

BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
	PAINTSTRUCT		ps;
	MEMORYSTATUS	memoryStatus;
	OSVERSIONINFO	VI;
	char			szTmp[256];
	static char		szDllPath[256];
	static char		szExePath[256];

	switch(uMsg)
	{
		case WM_INITDIALOG:
			GlobalMemoryStatus(&memoryStatus);
			iMaxPhysicalMemory	= (memoryStatus.dwTotalPhys/1048576);

			if ( GetInstallPath(szDllPath) )
			{
				MessageBox(hwnd, "Error internet service inetsrv is not installed.",
NULL, MB_ICONSTOP | MB_OK);
				EndDialog(hwnd, FALSE);
				return TRUE;
			}

			// set default values
			ZeroMemory( &Reg, sizeof(Reg) );
			Reg.dwNumberOfDeliveryThreads = 4;
			Reg.dwMaxConnections = 100;
			Reg.dwMaxPendingDeliveries	= 100;
			Reg.eDB_Protocol = DBLIB;
			Reg.eTxnMon = None;
			strcpy(Reg.szDbServer,		"");
			strcpy(Reg.szDbName,		"tpcc");
			strcpy(Reg.szDbUser,		"sa");
			strcpy(Reg.szDbPassword,	"");

			iPoolThreadLimit=  iMaxPhysicalMemory * 2;
			iThreadTimeout = 86400;
			iListenBackLog = 15;
			iAcceptExOutstanding = 40;

			ReadTPCCRegistrySettings( &Reg );
```

```
                ReadRegistrySettings();

                GetModuleFileName(hInst, szExePath, sizeof(szExePath));
                GetVersionInfo(szDllPath, szExePath);

                wsprintf(szTmp, "Version %d.%2.2d.%3.3d", versionExeMS, versionExeMM,
versionExeLS);
                SetDlgItemText(hwnd, IDC_VERSION, szTmp);

                SetDlgItemText(hwnd, IDC_PATH, szDllPath);

                SetDlgItemText(hwnd, ED_DB_SERVER, Reg.szDbServer);
                SetDlgItemText(hwnd, ED_DB_USER_ID, Reg.szDbUser);
                SetDlgItemText(hwnd, ED_DB_PASSWORD, Reg.szDbPassword);
                SetDlgItemText(hwnd, ED_DB_NAME, Reg.szDbName);

                SetDlgItemInt(hwnd, ED_THREADS, Reg.dwNumberOfDeliveryThreads, FALSE);
                SetDlgItemInt(hwnd, ED_MAXCONNECTION, Reg.dwMaxConnections, FALSE);
                SetDlgItemInt(hwnd, ED_MAXDELIVERIES, Reg.dwMaxPendingDeliveries,
FALSE);
                SetDlgItemInt(hwnd, ED_IIS_MAX_THEAD_POOL_LIMIT, iPoolThreadLimit,
FALSE);
                SetDlgItemInt(hwnd, ED_IIS_THREAD_TIMEOUT, iThreadTimeout, FALSE);
                SetDlgItemInt(hwnd, ED_IIS_LISTEN_BACKLOG, iListenBackLog, FALSE);
                SetDlgItemInt(hwnd, ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,
iAcceptExOutstanding, FALSE);

                CheckDlgButton(hwnd, IDC_DBLIB, 0);
                CheckDlgButton(hwnd, IDC_ODBC,  0);
                if ( Reg.eDB_Protocol == DBLIB )
                        CheckDlgButton(hwnd, IDC_DBLIB, 1);
                else
                        CheckDlgButton(hwnd, IDC_ODBC,  1);

                // check OS version level for COM.  Must be at least Windows 2000
                VI.dwOSVersionInfoSize = sizeof(VI);
                GetVersionEx( &VI );
                if (VI.dwMajorVersion < 5)
                {
                        HWND hDlg = GetDlgItem( hwnd, IDC_TM_MTS );
                        EnableWindow( hDlg, 0 );                // disable COM option
                        if (Reg.eTxnMon == COM)
                                Reg.eTxnMon = None;
                }

                CheckDlgButton(hwnd, IDC_TM_NONE,   0);
                CheckDlgButton(hwnd, IDC_TM_TUXEDO, 0);
                CheckDlgButton(hwnd, IDC_TM_MTS,    0);
                CheckDlgButton(hwnd, IDC_TM_ENCINA,         0);
                switch (Reg.eTxnMon)
                {
                case None:
                        CheckDlgButton(hwnd, IDC_TM_NONE,   1);
                        break;
                case TUXEDO:
                        CheckDlgButton(hwnd, IDC_TM_TUXEDO, 1);
                        break;
                case ENCINA:
                        CheckDlgButton(hwnd, IDC_TM_ENCINA, 1);
                        break;
                case COM:
                        CheckDlgButton(hwnd, IDC_TM_MTS,            1);
                        break;
                }

                return TRUE;
        case WM_PAINT:
                if ( IsIconic(hwnd) )
                {
                        BeginPaint(hwnd, &ps);
                        DrawIcon(ps.hdc, 0, 0, hIcon);
                        EndPaint(hwnd, &ps);
                        return TRUE;
                }
                break;
        case WM_COMMAND:
                if ( HIWORD(wParam) == BN_CLICKED )
                {
                        switch( LOWORD(wParam) )
                        {
                                case IDC_DBLIB:
                                        return TRUE;
                                case IDC_ODBC:
                                        return TRUE;
                                case IDOK:
                                        ProcessOK(hwnd, szDllPath);
                                        return TRUE;
                                case IDCANCEL:
                                        EndDialog(hwnd, FALSE);
                                        return TRUE;
                                default:
                                        return FALSE;
                        }
                }
                break;
        default:
                break;
        }
        return FALSE;
}

static void ProcessOK(HWND hwnd, char *szDllPath)
{
        int             d;
        HWND    hDlg;
        int             rc;

        char    szFullName[256];
        char    szErrTxt[128];

        // read settings from dialog
        Reg.dwNumberOfDeliveryThreads = GetDlgItemInt(hwnd, ED_THREADS, &d, FALSE);
```

```
		Reg.dwMaxConnections = GetDlgItemInt(hwnd, ED_MAXCONNECTION, &d, FALSE);
		Reg.dwMaxPendingDeliveries = GetDlgItemInt(hwnd, ED_MAXDELIVERIES, &d, FALSE);

		GetDlgItemText(hwnd, ED_DB_SERVER, Reg.szDbServer, sizeof(Reg.szDbServer));
		GetDlgItemText(hwnd, ED_DB_USER_ID, Reg.szDbUser, sizeof(Reg.szDbUser));
		GetDlgItemText(hwnd, ED_DB_PASSWORD, Reg.szDbPassword, sizeof(Reg.szDbPassword));
		GetDlgItemText(hwnd, ED_DB_NAME, Reg.szDbName, sizeof(Reg.szDbName));

		if ( IsDlgButtonChecked(hwnd, IDC_DBLIB) )
		{
			Reg.eDB_Protocol = DBLIB;
			rc = 1;
		}
		else if ( IsDlgButtonChecked(hwnd, IDC_ODBC) )
		{
			Reg.eDB_Protocol = ODBC;
			rc = 2;
		}

		if ( IsDlgButtonChecked(hwnd, IDC_TM_NONE) )
			Reg.eTxnMon = None;
		else if ( IsDlgButtonChecked(hwnd, IDC_TM_TUXEDO) )
			Reg.eTxnMon = TUXEDO;
		else if ( IsDlgButtonChecked(hwnd, IDC_TM_MTS) )
			Reg.eTxnMon = COM;
		else if ( IsDlgButtonChecked(hwnd, IDC_TM_ENCINA) )
			Reg.eTxnMon = ENCINA;

		iPoolThreadLimit = GetDlgItemInt(hwnd, ED_IIS_MAX_THEAD_POOL_LIMIT, &d, FALSE);
		iThreadTimeout = GetDlgItemInt(hwnd, ED_IIS_THREAD_TIMEOUT,	&d, FALSE);
		iListenBackLog = GetDlgItemInt(hwnd, ED_IIS_LISTEN_BACKLOG, &d, FALSE);
		iAcceptExOutstanding = GetDlgItemInt(hwnd, ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, &d,
FALSE);


		ShowWindow(hwnd, SW_HIDE);
		hDlg = CreateDialog(hInst, MAKEINTRESOURCE(IDD_DIALOG3), hwnd, CopyDlgProc);
		ShowWindow(hDlg, SW_SHOWNA);
		UpdateDialog(hDlg);


		// write binaries to inetpub\wwwroot
		rc = CopyFiles(hDlg, szDllPath);
		if ( !rc )
		{
			ShowWindow(hwnd, SW_SHOWNA);
			DestroyWindow(hDlg);
			strcpy( szErrTxt, "Error(s) occured when creating " );
			strcat( szErrTxt, szLastFileName );
			MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
			EndDialog(hwnd, 0);
			return;
		}

		// update registry
		SetDlgItemText(hDlg, IDC_STATUS, "Updating Registry.");
		SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
		UpdateDialog(hDlg);
		WriteRegistrySettings(szDllPath);

		// register com proxy stub
		strcpy(szFullName, szDllPath);
		strcat(szFullName, "tpcc_com_ps.dll");
		if (!RegisterDLL(szFullName))
		{
			ShowWindow(hwnd, SW_SHOWNA);
			DestroyWindow(hDlg);
			strcpy( szErrTxt, "Error occured when registering " );
			strcat( szErrTxt, szFullName );
			MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
			EndDialog(hwnd, 0);
			return;
		}

		// if using COM
		if (Reg.eTxnMon == COM)
		{
			SetDlgItemText(hDlg, IDC_STATUS, "Configuring COM.");
			SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
			UpdateDialog(hDlg);

			if (install_com(szDllPath))
			{
				ShowWindow(hwnd, SW_SHOWNA);
				DestroyWindow(hDlg);
				strcpy( szErrTxt, "Error occured when configuring COM settings." );
				MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
				EndDialog(hwnd, 0);
				return;
			}
		}

		Sleep(100);

		ShowWindow(hwnd, SW_SHOWNA);
		DestroyWindow(hDlg);

		EndDialog(hwnd, rc);
		return;
}

static void ReadRegistrySettings(void)
{
		HKEY	hKey;
		DWORD	size;
		DWORD	type;

		if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters",0, KEY_READ, &hKey) == ERROR_SUCCESS )
		{
			size = sizeof(iPoolThreadLimit);
```

```
                if ( RegQueryValueEx(hKey, "PoolThreadLimit", 0, &type, (char *)&iPoolThreadLimit, &size) ==
ERROR_SUCCESS )
                        if ( !iPoolThreadLimit)
                                iPoolThreadLimit = iMaxPhysicalMemory * 2;

                size = sizeof(iThreadTimeout);
                if ( RegQueryValueEx(hKey, "ThreadTimeout", 0, &type, (char *)&iThreadTimeout, &size) ==
ERROR_SUCCESS )
                        if ( !iThreadTimeout)
                                iThreadTimeout = 86400;

                size = sizeof(iListenBackLog);
                if ( RegQueryValueEx(hKey, "ListenBackLog", 0, &type, (char *)&iListenBackLog, &size) ==
ERROR_SUCCESS )
                        if ( !iListenBackLog)
                                iListenBackLog = 15;

                RegCloseKey(hKey);
        }

        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters",0, KEY_READ, &hKey) == ERROR_SUCCESS )
        {
                size = sizeof(iAcceptExOutstanding);
                if ( RegQueryValueEx(hKey, "AcceptExOutstanding", 0, &type, (char *)&iAcceptExOutstanding,
&size) == ERROR_SUCCESS )
                        if ( !iAcceptExOutstanding)
                                iAcceptExOutstanding = 40;

                RegCloseKey(hKey);
        }
}

static void WriteRegistrySettings(char *szDllPath)
{
        HKEY    hKey;
        DWORD   dwDisposition;
        char    szTmp[256];
        char    *ptr;
        int             iRc;

        if ( RegCreateKeyEx(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\TPCC", 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition) == ERROR_SUCCESS )
        {
                strcpy(szTmp, szDllPath);
                ptr = strstr(szTmp, "tpcc");
                if ( ptr )
                        *ptr = 0;

                RegSetValueEx(hKey, "Path", 0, REG_SZ, szTmp, strlen(szTmp)+1);

                RegSetValueEx(hKey, "NumberOfDeliveryThreads", 0, REG_DWORD, (char
*)&Reg.dwNumberOfDeliveryThreads, sizeof(Reg.dwNumberOfDeliveryThreads));
                RegSetValueEx(hKey, "MaxConnections", 0, REG_DWORD, (char *)&Reg.dwMaxConnections,
sizeof(Reg.dwMaxConnections));
                RegSetValueEx(hKey, "MaxPendingDeliveries", 0, REG_DWORD, (char
*)&Reg.dwMaxPendingDeliveries, sizeof(Reg.dwMaxPendingDeliveries));

                RegSetValueEx(hKey, "DB_Protocol", 0, REG_SZ, szDBNames[Reg.eDB_Protocol],
strlen(szDBNames[Reg.eDB_Protocol])+1);
                RegSetValueEx(hKey, "TxnMonitor", 0, REG_SZ, szTxnMonNames[Reg.eTxnMon],
strlen(szTxnMonNames[Reg.eTxnMon])+1);

                RegSetValueEx(hKey, "DbServer", 0, REG_SZ, Reg.szDbServer, strlen(Reg.szDbServer)+1);
                RegSetValueEx(hKey, "DbName", 0, REG_SZ, Reg.szDbName, strlen(Reg.szDbName)+1);
                RegSetValueEx(hKey, "DbUser", 0, REG_SZ, Reg.szDbUser, strlen(Reg.szDbUser)+1);
                RegSetValueEx(hKey, "DbPassword", 0, REG_SZ, Reg.szDbPassword,
strlen(Reg.szDbPassword)+1);

                strcpy(szTmp, "YES");
                RegSetValueEx(hKey, "COM_SinglePool", 0, REG_SZ, szTmp, strlen(szTmp)+1);

                RegFlushKey(hKey);
                RegCloseKey(hKey);
        }

        if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters",0, NULL, REG_OPTION_NON_VOLATILE,
KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) == ERROR_SUCCESS )
        {
                RegSetValueEx(hKey, "PoolThreadLimit", 0, REG_DWORD, (char *)&iPoolThreadLimit,
sizeof(iPoolThreadLimit));
                RegSetValueEx(hKey, "ThreadTimeout", 0, REG_DWORD, (char *)&iThreadTimeout,
sizeof(iThreadTimeout));
                RegSetValueEx(hKey, "ListenBackLog", 0, REG_DWORD, (char *)&iListenBackLog,
sizeof(iListenBackLog));

                RegFlushKey(hKey);
                RegCloseKey(hKey);
        }

        if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters",0, NULL, REG_OPTION_NON_VOLATILE,
KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) == ERROR_SUCCESS )
        {
                RegSetValueEx(hKey, "AcceptExOutstanding", 0, REG_DWORD, (char
*)&iAcceptExOutstanding, sizeof(iAcceptExOutstanding));

                RegFlushKey(hKey);
                RegCloseKey(hKey);
        }

        return;
}

BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
        if ( uMsg == WM_INITDIALOG )
        {
                SendDlgItemMessage(hwnd, IDC_PROGRESS1, PBM_SETRANGE, 0, MAKELPARAM(0,
15));
```

```
                SendDlgItemMessage(hwnd, IDC_PROGRESS1, PBM_SETSTEP, (WPARAM)1, 0);
                return TRUE;
        }
        return FALSE;
}

BOOL RegisterDLL(char *szFileName)
{
        HINSTANCE       hLib;
        FARPROC         lpDllEntryPoint;

        hLib = LoadLibrary(szFileName);
        if ( hLib == NULL )
                return FALSE;
        // Find the entry point.
        lpDllEntryPoint = GetProcAddress(hLib, "DllRegisterServer");
        if (lpDllEntryPoint != NULL)
        {
                return ((*lpDllEntryPoint)()== S_OK);
        }
        else
                return FALSE;           //unable to locate entry point
}

BOOL FileFromResource( char *szResourceName, int iResourceId, char *szDllPath, char *szFileName )
{
        HGLOBAL                 hDLL;
        HRSRC                   hResInfo;
        HANDLE                  hFile;
        DWORD                   dwSize;
        BYTE                    *pSrc;
        DWORD                   d;
        char                    szFullName[256];

        hResInfo = FindResource(hInst, MAKEINTRESOURCE(iResourceId), szResourceName);

        strcpy(szFullName, szDllPath);
        strcat(szFullName, szFileName);

        dwSize = SizeofResource(hInst, hResInfo);
        hDLL = LoadResource(hInst, hResInfo );
        pSrc = (BYTE *)LockResource(hDLL);
        remove(szFullName);

        if ( !(hFile = CreateFile(szFullName, GENERIC_WRITE, 0, NULL, CREATE_ALWAYS,
FILE_ATTRIBUTE_NORMAL, NULL)) )
                return FALSE;

        if ( !WriteFile(hFile,pSrc, dwSize, &d, NULL) )
                return FALSE;

        CloseHandle(hFile);

        UnlockResource(hDLL);
        FreeResource(hDLL);
        return TRUE;
}

static int CopyFiles(HWND hDlg, char *szDllPath)
{
        BOOL                    bSvcRunning;

        bSvcRunning = CheckWWWebService();
        if ( bSvcRunning )
        {
                SetDlgItemText(hDlg, IDC_STATUS, "Stopping Web Service.");
                SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
                UpdateDialog(hDlg);

                StopWWWebService();
                SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
                UpdateDialog(hDlg);
        }

        SetDlgItemText(hDlg, IDC_STATUS, "Copying Files...");
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install TPCC.DLL
        strcpy( szLastFileName, "tpcc.dll" );
        if (!FileFromResource( "TPCCDLL", IDR_TPCCDLL, szDllPath, szLastFileName ))
                return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_dblib.dll
        strcpy( szLastFileName, "tpcc_dblib.dll" );
        if (!FileFromResource( "DBLIB_DLL", IDR_DBLIB_DLL, szDllPath, szLastFileName ))
                return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_odbc.dll
        strcpy( szLastFileName, "tpcc_odbc.dll" );
        if (!FileFromResource( "ODBC_DLL", IDR_ODBC_DLL, szDllPath, szLastFileName ))
                return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tuxapp.exe
        strcpy( szLastFileName, "tuxapp.exe" );
        if (!FileFromResource( "TUXEDO_APP", IDR_TUXEDO_APP, szDllPath, szLastFileName ))
                return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_tuxedo.dll
        strcpy( szLastFileName, "tpcc_tuxedo.dll" );
        if (!FileFromResource( "TUXEDO_DLL", IDR_TUXEDO_DLL, szDllPath, szLastFileName ))
                return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);
```

```
		// install tpcc_com.dll
		strcpy( szLastFileName, "tpcc_com.dll" );
		if (!FileFromResource( "COM_DLL", IDR_COM_DLL, szDllPath, szLastFileName ))
			return 0;
		SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
		UpdateDialog(hDlg);

		// install tpcc_com_ps.dll
		strcpy( szLastFileName, "tpcc_com_ps.dll" );
		if (!FileFromResource( "COM_PS_DLL", IDR_COMPS_DLL, szDllPath, szLastFileName ))
			return 0;
		SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
		UpdateDialog(hDlg);

		// install tpcc_com_all.dll
		strcpy( szLastFileName, "tpcc_com_all.dll" );
		if (!FileFromResource( "COM_ALL_DLL", IDR_COMALL_DLL, szDllPath, szLastFileName ))
			return 0;
		SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
		UpdateDialog(hDlg);

		//if we stopped service restart it.
		if ( bSvcRunning )
		{
			SetDlgItemText(hDlg, IDC_STATUS, "Starting Web Service.");
			SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
			UpdateDialog(hDlg);
			StartWWWebService();
		}

		SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
		UpdateDialog(hDlg);

		return 1;
}

static BOOL GetInstallPath(char *szDllPath)
{
   HKEY	hKey;
		BYTE	szData[256];
		DWORD	sv;
		BOOL	bRc;
		int			len;
		char	*ptr;
		int			iRc;

		szDllPath[0] =  0;
		bRc = TRUE;
		if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\Virtual Roots", 0,KEY_ALL_ACCESS, &hKey) ==
ERROR_SUCCESS )
		{
			sv = sizeof(szData);
			iRc = RegQueryValueEx( hKey, "/,", NULL, NULL, szData, &sv );  // used by IIS 3.0
			if (iRc == ERROR_FILE_NOT_FOUND)
			{
				iRc = RegQueryValueEx( hKey, "/", NULL, NULL, szData, &sv );   // used by IIS
4.0
			if (iRc == ERROR_SUCCESS)
			{
				bRc = FALSE;
				strcpy(szDllPath, szData);
				if ( (ptr = strchr(szDllPath, ',')) )
					*ptr = 0;

				len = strlen(szDllPath);
				if ( szDllPath[len-1] != '\\' )
				{
					szDllPath[len] = '\\';
					szDllPath[len+1] = 0;
				}
			}

			RegCloseKey(hKey);
		}

		return bRc;
}

static void GetVersionInfo(char *szDLLPath, char *szExePath)
{
		DWORD				d;
		DWORD				dwSize;
		DWORD				dwBytes;
		char				*ptr;
		VS_FIXEDFILEINFO	*vs;

		versionDllMS = 0;
		versionDllLS = 0;
		if ( _access(szDLLPath, 00) == 0 )
		{
			dwSize = GetFileVersionInfoSize(szDLLPath,&d);
			if ( dwSize )
			{
				ptr = (char *)malloc(dwSize);
				GetFileVersionInfo(szDLLPath,0, dwSize, ptr);
				VerQueryValue(ptr, "\\",&vs, &dwBytes);
				versionDllMS = vs->dwProductVersionMS;
				versionDllLS = vs->dwProductVersionLS;
				free(ptr);
			}
		}

		versionExeMS = 0x7FFF;
		versionExeLS = 0x7FFF;
		dwSize = GetFileVersionInfoSize(szExePath,&d);
		if ( dwSize )
		{
			ptr = (char *)malloc(dwSize);
			GetFileVersionInfo(szExePath, 0, dwSize, ptr);
			VerQueryValue(ptr, "\\",&vs, &dwBytes);
```

```
                versionExeMS = vs->dwProductVersionMS;
                versionExeLS = LOWORD(vs->dwProductVersionLS);
                versionExeMM = HIWORD(vs->dwProductVersionLS);
                free(ptr);
        }
        return;
}

static BOOL CheckWWWebService(void)
{
        SC_HANDLE               schSCManager;
        SC_HANDLE               schService;
        SERVICE_STATUS      ssStatus;

        schSCManager = OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);
        schService = OpenService(schSCManager, TEXT("W3SVC"), SERVICE_ALL_ACCESS);
        if (schService == NULL)
                return FALSE;

        if (! QueryServiceStatus(schService, &ssStatus) )
                goto ServiceNotRunning;

        if ( !ControlService(schService, SERVICE_CONTROL_STOP, &ssStatus) )
                goto ServiceNotRunning;
        //start Service pending, Check the status until the service is running.
        if (! QueryServiceStatus(schService, &ssStatus) )
                goto ServiceNotRunning;

        CloseServiceHandle(schService);
        return TRUE;

ServiceNotRunning:

        CloseServiceHandle(schService);
        return FALSE;
}

static BOOL StartWWWebService(void)
{
        SC_HANDLE               schSCManager;
        SC_HANDLE               schService;
        SERVICE_STATUS      ssStatus;
        DWORD                   dwOldCheckPoint;

        schSCManager = OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);
        schService = OpenService(schSCManager, TEXT("W3SVC"), SERVICE_ALL_ACCESS);
        if (schService == NULL)
                return FALSE;

        if (! StartService(schService, 0, NULL) )
                goto StartWWWebErr;
        //start Service pending, Check the status until the service is running.
        if (! QueryServiceStatus(schService, &ssStatus) )
                goto StartWWWebErr;
        while( ssStatus.dwCurrentState != SERVICE_RUNNING)
        {
                dwOldCheckPoint = ssStatus.dwCheckPoint;                //Save the current
checkpoint.
                Sleep(ssStatus.dwWaitHint);
        //Wait for the specified interval.
                if ( !QueryServiceStatus(schService, &ssStatus) )    //Check the status again.
                        break;
                if (dwOldCheckPoint >= ssStatus.dwCheckPoint)                //Break if the checkpoint has not
been incremented.
                        break;
        }

        if (ssStatus.dwCurrentState == SERVICE_RUNNING)
                goto StartWWWebErr;

        CloseServiceHandle(schService);
        return TRUE;

StartWWWebErr:
        CloseServiceHandle(schService);
        return FALSE;
}

static BOOL StopWWWebService(void)
{
        SC_HANDLE               schSCManager;
        SC_HANDLE               schService;
        SERVICE_STATUS      ssStatus;
        DWORD                   dwOldCheckPoint;

        schSCManager = OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);
        schService = OpenService(schSCManager, TEXT("W3SVC"), SERVICE_ALL_ACCESS);
        if (schService == NULL)
                return FALSE;

        if (! QueryServiceStatus(schService, &ssStatus) )
                goto StopWWWebErr;

        if ( !ControlService(schService, SERVICE_CONTROL_STOP, &ssStatus) )
                goto StopWWWebErr;
        //start Service pending, Check the status until the service is running.
        if (! QueryServiceStatus(schService, &ssStatus) )
                goto StopWWWebErr;
        while( ssStatus.dwCurrentState == SERVICE_RUNNING)
        {
                dwOldCheckPoint = ssStatus.dwCheckPoint;                //Save the current
checkpoint.
                Sleep(ssStatus.dwWaitHint);
        //Wait for the specified interval.
                if ( !QueryServiceStatus(schService, &ssStatus) )    //Check the status again.
                        break;
                if (dwOldCheckPoint >= ssStatus.dwCheckPoint)                //Break if the checkpoint has not
been incremented.
                        break;
        }
```

```
        if (ssStatus.dwCurrentState == SERVICE_RUNNING)
                goto StopWWWebErr;

        CloseServiceHandle(schService);
        return TRUE;

StopWWWebErr:
        CloseServiceHandle(schService);
        return FALSE;
}

static void UpdateDialog(HWND hDlg)
{
        MSG msg;

        UpdateWindow(hDlg);
        while( PeekMessage(&msg, hDlg, 0, 0, PM_REMOVE) )
        {
                TranslateMessage(&msg);
                DispatchMessage(&msg);
        }
        Sleep(250);
        return;
}
```

install.dsp

```
# Microsoft Developer Studio Project File - Name="install" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=install - Win32 Release
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "install.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "install.mak" CFG="install - Win32 Release"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "install - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "install - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "install - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir ".\Release"
# PROP BASE Intermediate_Dir ".\Release"
# PROP BASE Target_Dir "."
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir "."
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386
# ADD LINK32 version.lib comctl32.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386 /out:"..\bin\install.exe"

!ELSEIF  "$(CFG)" == "install - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir "."
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir "."
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
```

```
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /debug /machine:I386
# ADD LINK32 version.lib comctl32.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /debug /machine:I386
/out:"..\bin\install.exe"

!ENDIF

# Begin Target

# Name "install - Win32 Release"
# Name "install - Win32 Debug"
# Begin Group "Source Files"

# PROP Default_Filter "cpp;c;cxx;rc;def;r;odl;hpj;bat;for;f90"
# Begin Source File

SOURCE=.\src\install.c
# End Source File
# Begin Source File

SOURCE=.\src\install.rc
# ADD BASE RSC /l 0x409 /i "src"
# ADD RSC /l 0x409 /i "src" /i "..\src"
# End Source File
# Begin Source File

SOURCE=.\src\install_com.cpp
# End Source File
# End Group
# Begin Group "Header Files"

# PROP Default_Filter "h;hpp;hxx;hm;inl;fi;fd"
# End Group
# Begin Group "Resource Files"

# PROP Default_Filter "ico;cur;bmp;dlg;rc2;rct;bin;cnt;rtf;gif;jpg;jpeg;jpe"
# Begin Source File

SOURCE=.\SRC\ICON1.ICO
# End Source File
# Begin Source File

SOURCE=.\SRC\ICON2.ICO
# End Source File
# End Group
# Begin Source File

SOURCE=.\SRC\LICENSE.TXT
# End Source File
# Begin Source File

SOURCE=..\isapi_dll\bin\tpcc.dll
# End Source File
# Begin Source File

SOURCE=..\tm_com_dll\bin\tpcc_com.dll
# End Source File
# Begin Source File

SOURCE=..\tpcc_com_all\bin\tpcc_com_all.dll
# End Source File
# Begin Source File

SOURCE=..\tpcc_com_ps\bin\tpcc_com_ps.dll
# End Source File
# Begin Source File

SOURCE=..\db_dblib_dll\bin\tpcc_dblib.dll
# End Source File
# Begin Source File

SOURCE=..\db_odbc_dll\bin\tpcc_odbc.dll
# End Source File
# Begin Source File

SOURCE=..\tm_tuxedo_dll\bin\tpcc_tuxedo.dll
# End Source File
# Begin Source File

SOURCE=..\tuxapp\bin\tuxapp.exe
# End Source File
# End Target
# End Project
```

install.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//

#define IDD_DIALOG1                     101
#define IDI_ICON1                       102
#define IDR_TPCCDLL                     103
#define IDD_DIALOG2                     105
#define IDI_ICON2                       106
#define IDR_DELIVERY                    107
#define IDD_DIALOG3                     108

#define BN_LOG                          1001
#define ED_KEEP                         1002
#define ED_THREADS                      1003
#define ED_THREADS2                     1004
#define IDC_PATH                        1007
#define IDC_VERSION                     1009
```

```
#define IDC_RESULTS             1010
#define IDC_PROGRESS1                   1011
#define IDC_STATUS              1012
#define IDC_BUTTON1             1013
#define ED_MAXCONNECTION                1014
#define ED_IIS_MAX_THEAD_POOL_LIMIT     1015
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE       1017
#define ED_IIS_THREAD_TIMEOUT           1018
#define ED_IIS_LISTEN_BACKLOG           1019
#define IDC_DBLIB               1021
#define IDC_ODBC                1022
#define IDC_CONNECT_POOL                                                1023
#define ED_USER_CONNECT_DELAY_TIME                              1024

// Next default values for new objects
//
```

install.rc

```
/*      FILE:           INSTALL.C
 *                              Microsoft TPC-C Kit Ver. 4.20.000
 *                              Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                              not audited
 *
 *      PURPOSE:  Automated installation application forTPC-C Web Kit
 *      Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                  4.20.000 - added COM installation steps
 */

#include <windows.h>
#include <direct.h>
#include <io.h>
#include <stdlib.h>
#include <stdio.h>
#include <commctrl.h>
#include "..\..\common\src\ReadRegistry.h"

#include "resource.h"

#define WM_INITTEXT             WM_USER+100

HICON           hIcon;
HINSTANCE       hInst;

DWORD                   versionExeMS;
DWORD                   versionExeLS;
DWORD                   versionExeMM;
DWORD                   versionDllMS;
DWORD                   versionDllLS;

// TPC-C registry settings
TPCCREGISTRYDATA Reg;

static  int             iPoolThreadLimit;
static  int             iThreadTimeout;
static  int             iListenBackLog;
static  int             iAcceptExOutstanding;

static  int             iMaxPhysicalMemory;     //max physical memory in MB
static  char    szLastFileName[64];     // last file we worked on (for error reporting)


BOOL    CALLBACK        LicenseDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam);
BOOL    CALLBACK        UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam);
BOOL    CALLBACK        MainDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
BOOL    CALLBACK        CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
static  void            ProcessOK(HWND hwnd, char *szDllPath);
static  void            ReadRegistrySettings(void);
static  void            WriteRegistrySettings(char *szDllPath);
static  BOOL            RegisterDLL(char *szFileName);
static  int                     CopyFiles(HWND hDlg, char *szDllPath);
static  BOOL            GetInstallPath(char *szDllPath);
static  void            GetVersionInfo(char *szDLLPath, char *szExePath);
static  BOOL            CheckWWWebService(void);
static  BOOL            StartWWWebService(void);
static  BOOL            StopWWWebService(void);
static  void            UpdateDialog(HWND hDlg);

BOOL install_com(char *szDllPath);

#include "..\..\common\src\ReadRegistry.cpp"

int WINAPI WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow )
{
        int iRc;

        hInst = hInstance;

        InitCommonControls();

        hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_ICON1));

        iRc = DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG4), GetDesktopWindow(),
LicenseDlgProc);
        if ( iRc )
        {
                iRc = DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG1), GetDesktopWindow(),
MainDlgProc);
                if ( iRc )
                {
                        DialogBoxParam(hInstance, MAKEINTRESOURCE(IDD_DIALOG2),
GetDesktopWindow(), UpdatedDlgProc, (LPARAM)iRc);
                }
        }
```

```
	DestroyIcon(hIcon);
	return 0;
}

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
	HGLOBAL			hRes;
	HRSRC			hResInfo;
	BYTE			*pSrc, *pDst;
	DWORD			dwSize;
	static HFONT		hFont;

	switch(uMsg)
	{
		case WM_INITDIALOG:
			hFont = CreateFont(-12, 0, 0, 0, 400, 0, 0, 0, 0, 0, 0, 0, 0,  "Arial");
			SendMessage( GetDlgItem(hwnd, IDR_LICENSE1), WM_SETFONT,
(WPARAM)hFont, MAKELPARAM(0, 0) );
			PostMessage(hwnd, WM_INITTEXT, (WPARAM)0, (LPARAM)0);
			return TRUE;
		case WM_INITTEXT:
			hResInfo = FindResource(hInst, MAKEINTRESOURCE(IDR_LICENSE1),
"LICENSE");
			dwSize = SizeofResource(hInst, hResInfo);
			hRes = LoadResource(hInst, hResInfo );
			pSrc = (BYTE *)LockResource(hRes);
			pDst = (unsigned char *)malloc(dwSize+1);
			if ( pDst )
			{
				memcpy(pDst, pSrc, dwSize);
				pDst[dwSize] = 0;
				SetDlgItemText(hwnd, IDC_LICENSE, (const char *)pDst);
				free(pDst);
			}
			else
				SetDlgItemText(hwnd, IDC_LICENSE, (const char *)pSrc);
			return TRUE;
		case WM_DESTROY:
			DeleteObject(hFont);
			return TRUE;
		case WM_COMMAND:
			if ( wParam == IDOK )
				EndDialog(hwnd, TRUE);
			if ( wParam == IDCANCEL )
				EndDialog(hwnd, FALSE);
		default:
			break;
	}
	return FALSE;
}

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
	switch(uMsg)
	{
		case WM_INITDIALOG:
			switch(lParam)
			{
				case 1:
				case 2:
					SetDlgItemText(hwnd, IDC_RESULTS, "TPC-C Web
Client Installed");
					break;
			}
			return TRUE;
		case WM_COMMAND:
			if ( wParam == IDOK )
				EndDialog(hwnd, TRUE);
			break;
		default:
			break;
	}
	return FALSE;
}

BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
	PAINTSTRUCT		ps;
	MEMORYSTATUS	memoryStatus;
	OSVERSIONINFO	VI;
	char			szTmp[256];
	static char		szDllPath[256];
	static char		szExePath[256];

	switch(uMsg)
	{
		case WM_INITDIALOG:
			GlobalMemoryStatus(&memoryStatus);
			iMaxPhysicalMemory	= (memoryStatus.dwTotalPhys/1048576);

			if ( GetInstallPath(szDllPath) )
			{
				MessageBox(hwnd, "Error internet service inetsrv is not installed.",
NULL, MB_ICONSTOP | MB_OK);
				EndDialog(hwnd, FALSE);
				return TRUE;
			}

			// set default values
			ZeroMemory( &Reg, sizeof(Reg) );
			Reg.dwNumberOfDeliveryThreads = 4;
			Reg.dwMaxConnections = 100;
			Reg.dwMaxPendingDeliveries	= 100;
			Reg.eDB_Protocol = DBLIB;
			Reg.eTxnMon = None;
			strcpy(Reg.szDbServer,		"");
			strcpy(Reg.szDbName,		"tpcc");
			strcpy(Reg.szDbUser,		"sa");
			strcpy(Reg.szDbPassword,	"");

			iPoolThreadLimit=  iMaxPhysicalMemory * 2;
```

```
                iThreadTimeout = 86400;
                iListenBackLog = 15;
                iAcceptExOutstanding = 40;

                ReadTPCCRegistrySettings( &Reg );
                ReadRegistrySettings();

                GetModuleFileName(hInst, szExePath, sizeof(szExePath));
                GetVersionInfo(szDllPath, szExePath);

                wsprintf(szTmp, "Version %d.%2.2d.%3.3d", versionExeMS, versionExeMM,
versionExeLS);
                SetDlgItemText(hwnd, IDC_VERSION, szTmp);

                SetDlgItemText(hwnd, IDC_PATH, szDllPath);

                SetDlgItemText(hwnd, ED_DB_SERVER, Reg.szDbServer);
                SetDlgItemText(hwnd, ED_DB_USER_ID, Reg.szDbUser);
                SetDlgItemText(hwnd, ED_DB_PASSWORD, Reg.szDbPassword);
                SetDlgItemText(hwnd, ED_DB_NAME, Reg.szDbName);

                SetDlgItemInt(hwnd, ED_THREADS, Reg.dwNumberOfDeliveryThreads, FALSE);
                SetDlgItemInt(hwnd, ED_MAXCONNECTION, Reg.dwMaxConnections, FALSE);
                SetDlgItemInt(hwnd, ED_MAXDELIVERIES, Reg.dwMaxPendingDeliveries,
FALSE);
                SetDlgItemInt(hwnd, ED_IIS_MAX_THEAD_POOL_LIMIT, iPoolThreadLimit,
FALSE);
                SetDlgItemInt(hwnd, ED_IIS_THREAD_TIMEOUT, iThreadTimeout, FALSE);
                SetDlgItemInt(hwnd, ED_IIS_LISTEN_BACKLOG, iListenBackLog, FALSE);
                SetDlgItemInt(hwnd, ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,
iAcceptExOutstanding, FALSE);

                CheckDlgButton(hwnd, IDC_DBLIB, 0);
                CheckDlgButton(hwnd, IDC_ODBC,  0);
                if ( Reg.eDB_Protocol == DBLIB )
                        CheckDlgButton(hwnd, IDC_DBLIB, 1);
                else
                        CheckDlgButton(hwnd, IDC_ODBC,  1);

                // check OS version level for COM.  Must be at least Windows 2000
                VI.dwOSVersionInfoSize = sizeof(VI);
                GetVersionEx( &VI );
                if (VI.dwMajorVersion < 5)
                {
                        HWND hDlg = GetDlgItem( hwnd, IDC_TM_MTS );
                        EnableWindow( hDlg, 0 );                // disable COM option
                        if (Reg.eTxnMon == COM)
                                Reg.eTxnMon = None;
                }

                CheckDlgButton(hwnd, IDC_TM_NONE,   0);
                CheckDlgButton(hwnd, IDC_TM_TUXEDO, 0);
                CheckDlgButton(hwnd, IDC_TM_MTS,    0);
                CheckDlgButton(hwnd, IDC_TM_ENCINA,         0);
                switch (Reg.eTxnMon)
                {
                        case None:
                                CheckDlgButton(hwnd, IDC_TM_NONE,   1);
                                break;
                        case TUXEDO:
                                CheckDlgButton(hwnd, IDC_TM_TUXEDO, 1);
                                break;
                        case ENCINA:
                                CheckDlgButton(hwnd, IDC_TM_ENCINA, 1);
                                break;
                        case COM:
                                CheckDlgButton(hwnd, IDC_TM_MTS,                1);
                                break;
                        }

                        return TRUE;
                case WM_PAINT:
                        if ( IsIconic(hwnd) )
                        {
                                BeginPaint(hwnd, &ps);
                                DrawIcon(ps.hdc, 0, 0, hIcon);
                                EndPaint(hwnd, &ps);
                                return TRUE;
                        }
                        break;
                case WM_COMMAND:
                        if ( HIWORD(wParam) == BN_CLICKED )
                        {
                                switch( LOWORD(wParam) )
                                {
                                        case IDC_DBLIB:
                                                return TRUE;
                                        case IDC_ODBC:
                                                return TRUE;
                                        case IDOK:
                                                ProcessOK(hwnd, szDllPath);
                                                return TRUE;
                                        case IDCANCEL:
                                                EndDialog(hwnd, FALSE);
                                                return TRUE;
                                        default:
                                                return FALSE;
                                }
                        }
                        break;
                default:
                        break;
        }
        return FALSE;
}

static void ProcessOK(HWND hwnd, char *szDllPath)
{
        int             d;
        HWND    hDlg;
        int             rc;
```

```
	char		szFullName[256];
	char		szErrTxt[128];

	// read settings from dialog
	Reg.dwNumberOfDeliveryThreads = GetDlgItemInt(hwnd, ED_THREADS, &d, FALSE);
	Reg.dwMaxConnections = GetDlgItemInt(hwnd, ED_MAXCONNECTION, &d, FALSE);
	Reg.dwMaxPendingDeliveries = GetDlgItemInt(hwnd, ED_MAXDELIVERIES, &d, FALSE);

	GetDlgItemText(hwnd, ED_DB_SERVER, Reg.szDbServer, sizeof(Reg.szDbServer));
	GetDlgItemText(hwnd, ED_DB_USER_ID, Reg.szDbUser, sizeof(Reg.szDbUser));
	GetDlgItemText(hwnd, ED_DB_PASSWORD, Reg.szDbPassword, sizeof(Reg.szDbPassword));
	GetDlgItemText(hwnd, ED_DB_NAME, Reg.szDbName, sizeof(Reg.szDbName));

	if ( IsDlgButtonChecked(hwnd, IDC_DBLIB) )
	{
		Reg.eDB_Protocol = DBLIB;
		rc = 1;
	}
	else if ( IsDlgButtonChecked(hwnd, IDC_ODBC) )
	{
		Reg.eDB_Protocol = ODBC;
		rc = 2;
	}

	if ( IsDlgButtonChecked(hwnd, IDC_TM_NONE) )
		Reg.eTxnMon = None;
	else if ( IsDlgButtonChecked(hwnd, IDC_TM_TUXEDO) )
		Reg.eTxnMon = TUXEDO;
	else if ( IsDlgButtonChecked(hwnd, IDC_TM_MTS) )
		Reg.eTxnMon = COM;
	else if ( IsDlgButtonChecked(hwnd, IDC_TM_ENCINA) )
		Reg.eTxnMon = ENCINA;

	iPoolThreadLimit = GetDlgItemInt(hwnd, ED_IIS_MAX_THEAD_POOL_LIMIT, &d, FALSE);
	iThreadTimeout = GetDlgItemInt(hwnd, ED_IIS_THREAD_TIMEOUT,	&d, FALSE);
	iListenBackLog = GetDlgItemInt(hwnd, ED_IIS_LISTEN_BACKLOG, &d, FALSE);
	iAcceptExOutstanding = GetDlgItemInt(hwnd, ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, &d,
FALSE);


	ShowWindow(hwnd, SW_HIDE);
	hDlg = CreateDialog(hInst, MAKEINTRESOURCE(IDD_DIALOG3), hwnd, CopyDlgProc);
	ShowWindow(hDlg, SW_SHOWNA);
	UpdateDialog(hDlg);


	// write binaries to inetpub\wwwroot
	rc = CopyFiles(hDlg, szDllPath);
	if ( !rc )
	{
		ShowWindow(hwnd, SW_SHOWNA);
		DestroyWindow(hDlg);
		strcpy( szErrTxt, "Error(s) occured when creating " );
		strcat( szErrTxt, szLastFileName );
		MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
		EndDialog(hwnd, 0);
		return;
	}

	// update registry
	SetDlgItemText(hDlg, IDC_STATUS, "Updating Registry.");
	SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
	UpdateDialog(hDlg);
	WriteRegistrySettings(szDllPath);

	// register com proxy stub
	strcpy(szFullName, szDllPath);
	strcat(szFullName, "tpcc_com_ps.dll");
	if (!RegisterDLL(szFullName))
	{
		ShowWindow(hwnd, SW_SHOWNA);
		DestroyWindow(hDlg);
		strcpy( szErrTxt, "Error occured when registering " );
		strcat( szErrTxt, szFullName );
		MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
		EndDialog(hwnd, 0);
		return;
	}

	// if using COM
	if (Reg.eTxnMon == COM)
	{
		SetDlgItemText(hDlg, IDC_STATUS, "Configuring COM.");
		SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
		UpdateDialog(hDlg);

		if (install_com(szDllPath))
		{
			ShowWindow(hwnd, SW_SHOWNA);
			DestroyWindow(hDlg);
			strcpy( szErrTxt, "Error occured when configuring COM settings." );
			MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
			EndDialog(hwnd, 0);
			return;
		}
	}

	Sleep(100);

	ShowWindow(hwnd, SW_SHOWNA);
	DestroyWindow(hDlg);

	EndDialog(hwnd, rc);
	return;
}

static void ReadRegistrySettings(void)
{
	HKEY	hKey;
	DWORD	size;
	DWORD	type;
```

```
	if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters",0, KEY_READ, &hKey) == ERROR_SUCCESS )
	{
		size = sizeof(iPoolThreadLimit);
		if ( RegQueryValueEx(hKey, "PoolThreadLimit", 0, &type, (char *)&iPoolThreadLimit,&size) ==
ERROR_SUCCESS )
			if ( !iPoolThreadLimit)
				iPoolThreadLimit= iMaxPhysicalMemory* 2;

		size = sizeof(iThreadTimeout);
		if ( RegQueryValueEx(hKey, "ThreadTimeout", 0, &type, (char *)&iThreadTimeout,&size) ==
ERROR_SUCCESS )
			if ( !iThreadTimeout)
				iThreadTimeout= 86400;

		size = sizeof(iListenBackLog);
		if ( RegQueryValueEx(hKey, "ListenBackLog", 0, &type, (char *)&iListenBackLog,&size) ==
ERROR_SUCCESS )
			if ( !iListenBackLog)
				iListenBackLog= 15;

		RegCloseKey(hKey);
	}

	if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters",0, KEY_READ, &hKey) == ERROR_SUCCESS )
	{
		size = sizeof(iAcceptExOutstanding);
		if ( RegQueryValueEx(hKey, "AcceptExOutstanding", 0, &type, (char *)&iAcceptExOutstanding,
&size) == ERROR_SUCCESS )
			if ( !iAcceptExOutstanding)
				iAcceptExOutstanding= 40;

		RegCloseKey(hKey);
	}
}

static void WriteRegistrySettings(char*szDllPath)
{
	HKEY	hKey;
	DWORD	dwDisposition;
	char	szTmp[256];
	char	*ptr;
	int		iRc;

	if ( RegCreateKeyEx(HKEY_LOCAL_MACHINE,"SOFTWARE\\Microsoft\\TPCC",0, NULL,
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition) == ERROR_SUCCESS )
	{
		strcpy(szTmp, szDllPath);
		ptr = strstr(szTmp, "tpcc");
		if ( ptr )
			*ptr = 0;

		RegSetValueEx(hKey, "Path", 0, REG_SZ, szTmp, strlen(szTmp)+1);

		RegSetValueEx(hKey, "NumberOfDeliveryThreads", 0, REG_DWORD, (char
*)&Reg.dwNumberOfDeliveryThreads,sizeof(Reg.dwNumberOfDeliveryThreads));
		RegSetValueEx(hKey, "MaxConnections", 0, REG_DWORD, (char *)&Reg.dwMaxConnections,
sizeof(Reg.dwMaxConnections));
		RegSetValueEx(hKey, "MaxPendingDeliveries", 0, REG_DWORD, (char
*)&Reg.dwMaxPendingDeliveries, sizeof(Reg.dwMaxPendingDeliveries));

		RegSetValueEx(hKey, "DB_Protocol", 0, REG_SZ, szDBNames[Reg.eDB_Protocol],
strlen(szDBNames[Reg.eDB_Protocol])+1);
		RegSetValueEx(hKey, "TxnMonitor", 0, REG_SZ, szTxnMonNames[Reg.eTxnMon],
strlen(szTxnMonNames[Reg.eTxnMon])+1);

		RegSetValueEx(hKey, "DbServer", 0, REG_SZ, Reg.szDbServer, strlen(Reg.szDbServer)+1);
		RegSetValueEx(hKey, "DbName", 0, REG_SZ, Reg.szDbName, strlen(Reg.szDbName)+1);
		RegSetValueEx(hKey, "DbUser", 0, REG_SZ, Reg.szDbUser, strlen(Reg.szDbUser)+1);
		RegSetValueEx(hKey, "DbPassword", 0, REG_SZ, Reg.szDbPassword,
strlen(Reg.szDbPassword)+1);

		strcpy(szTmp, "YES");
		RegSetValueEx(hKey, "COM_SinglePool", 0, REG_SZ, szTmp, strlen(szTmp)+1);

		RegFlushKey(hKey);
		RegCloseKey(hKey);
	}

	if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters",0, NULL, REG_OPTION_NON_VOLATILE,
KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) == ERROR_SUCCESS )
	{
		RegSetValueEx(hKey, "PoolThreadLimit", 0, REG_DWORD, (char *)&iPoolThreadLimit,
sizeof(iPoolThreadLimit));
		RegSetValueEx(hKey, "ThreadTimeout", 0, REG_DWORD, (char *)&iThreadTimeout,
sizeof(iThreadTimeout));
		RegSetValueEx(hKey, "ListenBackLog", 0, REG_DWORD, (char *)&iListenBackLog,
sizeof(iListenBackLog));

		RegFlushKey(hKey);
		RegCloseKey(hKey);
	}

	if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters",0, NULL, REG_OPTION_NON_VOLATILE,
KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) == ERROR_SUCCESS )
	{
		RegSetValueEx(hKey, "AcceptExOutstanding", 0, REG_DWORD, (char
*)&iAcceptExOutstanding, sizeof(iAcceptExOutstanding));

		RegFlushKey(hKey);
		RegCloseKey(hKey);
	}

	return;
}

BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
```

```
	if ( uMsg == WM_INITDIALOG )
	{
		SendDlgItemMessage(hwnd, IDC_PROGRESS1, PBM_SETRANGE, 0, MAKELPARAM(0,
15));
		SendDlgItemMessage(hwnd, IDC_PROGRESS1, PBM_SETSTEP, (WPARAM)1, 0);
		return TRUE;
	}
	return FALSE;
}

BOOL RegisterDLL(char *szFileName)
{
	HINSTANCE		hLib;
	FARPROC		lpDllEntryPoint;

	hLib = LoadLibrary(szFileName);
	if ( hLib == NULL )
		return FALSE;
	// Find the entry point.
	lpDllEntryPoint= GetProcAddress(hLib, "DllRegisterServer");
	if (lpDllEntryPoint != NULL)
	{
		return ((*lpDllEntryPoint)()== S_OK);
	}
	else
		return FALSE;		//unable to locate entry point
}

BOOL FileFromResource( char *szResourceName, int iResourceId, char *szDllPath, char *szFileName )
{
	HGLOBAL				hDLL;
	HRSRC				hResInfo;
	HANDLE				hFile;
	DWORD				dwSize;
	BYTE				*pSrc;
	DWORD				d;
	char				szFullName[256];

	hResInfo = FindResource(hInst, MAKEINTRESOURCE(iResourceId), szResourceName);

	strcpy(szFullName, szDllPath);
	strcat(szFullName, szFileName);

	dwSize = SizeofResource(hInst, hResInfo);
	hDLL = LoadResource(hInst, hResInfo );
	pSrc = (BYTE *)LockResource(hDLL);
	remove(szFullName);

	if ( !(hFile = CreateFile(szFullName, GENERIC_WRITE, 0, NULL, CREATE_ALWAYS,
FILE_ATTRIBUTE_NORMAL, NULL)) )
		return FALSE;

	if ( !WriteFile(hFile,pSrc, dwSize, &d, NULL) )
		return FALSE;

	CloseHandle(hFile);
	UnlockResource(hDLL);
	FreeResource(hDLL);
	return TRUE;
}

static int CopyFiles(HWND hDlg, char *szDllPath)
{
	BOOL			bSvcRunning;

	bSvcRunning = CheckWWWebService();
	if ( bSvcRunning )
	{
		SetDlgItemText(hDlg, IDC_STATUS, "Stopping Web Service.");
		SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
		UpdateDialog(hDlg);

		StopWWWebService();
		SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
		UpdateDialog(hDlg);
	}

	SetDlgItemText(hDlg, IDC_STATUS, "Copying Files...");
	SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
	UpdateDialog(hDlg);

	// install TPCC.DLL
	strcpy( szLastFileName, "tpcc.dll" );
	if (!FileFromResource( "TPCCDLL", IDR_TPCCDLL, szDllPath, szLastFileName ))
		return 0;
	SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
	UpdateDialog(hDlg);

	// install tpcc_dblib.dll
	strcpy( szLastFileName, "tpcc_dblib.dll" );
	if (!FileFromResource( "DBLIB_DLL", IDR_DBLIB_DLL, szDllPath, szLastFileName ))
		return 0;
	SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
	UpdateDialog(hDlg);

	// install tpcc_odbc.dll
	strcpy( szLastFileName, "tpcc_odbc.dll" );
	if (!FileFromResource( "ODBC_DLL", IDR_ODBC_DLL, szDllPath, szLastFileName ))
		return 0;
	SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
	UpdateDialog(hDlg);

	// install tuxapp.exe
	strcpy( szLastFileName, "tuxapp.exe" );
	if (!FileFromResource( "TUXEDO_APP", IDR_TUXEDO_APP, szDllPath, szLastFileName ))
		return 0;
	SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
	UpdateDialog(hDlg);

	// install tpcc_tuxedo.dll
	strcpy( szLastFileName, "tpcc_tuxedo.dll" );
```

```
        if (!FileFromResource( "TUXEDO_DLL", IDR_TUXEDO_DLL, szDllPath, szLastFileName ))
                return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_com.dll
        strcpy( szLastFileName, "tpcc_com.dll" );
        if (!FileFromResource( "COM_DLL", IDR_COM_DLL, szDllPath, szLastFileName ))
                return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_com_ps.dll
        strcpy( szLastFileName, "tpcc_com_ps.dll" );
        if (!FileFromResource( "COM_PS_DLL", IDR_COMPS_DLL, szDllPath, szLastFileName ))
                return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_com_all.dll
        strcpy( szLastFileName, "tpcc_com_all.dll" );
        if (!FileFromResource( "COM_ALL_DLL", IDR_COMALL_DLL, szDllPath, szLastFileName ))
                return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        //if we stopped service restart it.
        if ( bSvcRunning )
        {
                SetDlgItemText(hDlg, IDC_STATUS, "Starting Web Service.");
                SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
                UpdateDialog(hDlg);
                StartWWWebService();
        }

        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        return 1;
}

static BOOL GetInstallPath(char *szDllPath)
{
   HKEY    hKey;
        BYTE    szData[256];
        DWORD   sv;
        BOOL    bRc;
        int             len;
        char    *ptr;
        int             iRc;

        szDllPath[0] =  0;
        bRc = TRUE;
        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\Virtual Roots", 0,KEY_ALL_ACCESS, &hKey) ==
ERROR_SUCCESS )
        {
                sv = sizeof(szData);
                iRc = RegQueryValueEx( hKey, "/,", NULL, NULL, szData, &sv );  // used by IIS 3.0
                if (iRc == ERROR_FILE_NOT_FOUND)
                        iRc = RegQueryValueEx( hKey, "/", NULL, NULL, szData, &sv );   // used by IIS
4.0
                if (iRc == ERROR_SUCCESS)
                {
                        bRc = FALSE;
                        strcpy(szDllPath, szData);
                        if ( (ptr = strchr(szDllPath, ',')) )
                                *ptr = 0;

                        len = strlen(szDllPath);
                        if ( szDllPath[len-1] != '\\' )
                        {
                                szDllPath[len] = '\\';
                                szDllPath[len+1] = 0;
                        }
                }

                RegCloseKey(hKey);
        }

        return bRc;
}

static void GetVersionInfo(char *szDLLPath, char *szExePath)
{
        DWORD                                   d;
        DWORD                                   dwSize;
        DWORD                                   dwBytes;
        char                                    *ptr;
        VS_FIXEDFILEINFO    *vs;

        versionDllMS = 0;
        versionDllLS = 0;
        if ( _access(szDLLPath, 00) == 0 )
        {
                dwSize = GetFileVersionInfoSize(szDLLPath,&d);
                if ( dwSize )
                {
                        ptr = (char *)malloc(dwSize);
                        GetFileVersionInfo(szDLLPath,0, dwSize, ptr);
                        VerQueryValue(ptr, "\\",&vs, &dwBytes);
                        versionDllMS = vs->dwProductVersionMS;
                        versionDllLS = vs->dwProductVersionLS;
                        free(ptr);
                }
        }

        versionExeMS = 0x7FFF;
        versionExeLS = 0x7FFF;
        dwSize = GetFileVersionInfoSize(szExePath,&d);
        if ( dwSize )
        {
```

```
                ptr = (char *)malloc(dwSize);
                GetFileVersionInfo(szExePath, 0, dwSize, ptr);
                VerQueryValue(ptr, "\\",&vs, &dwBytes);

                versionExeMS = vs->dwProductVersionMS;
                versionExeLS = LOWORD(vs->dwProductVersionLS);
                versionExeMM = HIWORD(vs->dwProductVersionLS);
                free(ptr);
        }
        return;
}

static BOOL CheckWWWebService(void)
{
        SC_HANDLE               schSCManager;
        SC_HANDLE               schService;
        SERVICE_STATUS      ssStatus;

        schSCManager = OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);
        schService = OpenService(schSCManager, TEXT("W3SVC"), SERVICE_ALL_ACCESS);
        if (schService == NULL)
                return FALSE;

        if (! QueryServiceStatus(schService, &ssStatus) )
                goto ServiceNotRunning;

        if ( !ControlService(schService, SERVICE_CONTROL_STOP, &ssStatus) )
                goto ServiceNotRunning;
        //start Service pending, Check the status until the service is running.
        if (! QueryServiceStatus(schService, &ssStatus) )
                goto ServiceNotRunning;

        CloseServiceHandle(schService);
        return TRUE;

ServiceNotRunning:

        CloseServiceHandle(schService);
        return FALSE;
}

static BOOL StartWWWebService(void)
{
        SC_HANDLE               schSCManager;
        SC_HANDLE               schService;
        SERVICE_STATUS      ssStatus;
        DWORD                   dwOldCheckPoint;

        schSCManager = OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);
        schService = OpenService(schSCManager, TEXT("W3SVC"), SERVICE_ALL_ACCESS);
        if (schService == NULL)
                return FALSE;

        if (! StartService(schService, 0, NULL) )
                goto StartWWWebErr;
        //start Service pending, Check the status until the service is running.
        if (! QueryServiceStatus(schService, &ssStatus) )
                goto StartWWWebErr;
        while( ssStatus.dwCurrentState != SERVICE_RUNNING)
        {

                dwOldCheckPoint = ssStatus.dwCheckPoint;                        //Save the current
checkpoint.
                Sleep(ssStatus.dwWaitHint);
        //Wait for the specified interval.
                if ( !QueryServiceStatus(schService, &ssStatus) )    //Check the status again.
                        break;
                if (dwOldCheckPoint >= ssStatus.dwCheckPoint)                   //Break if the checkpoint has not
been incremented.
                        break;
        }

        if (ssStatus.dwCurrentState == SERVICE_RUNNING)
                goto StartWWWebErr;

        CloseServiceHandle(schService);
        return TRUE;

StartWWWebErr:
        CloseServiceHandle(schService);
        return FALSE;
}

static BOOL StopWWWebService(void)
{
        SC_HANDLE               schSCManager;
        SC_HANDLE               schService;
        SERVICE_STATUS      ssStatus;
        DWORD                   dwOldCheckPoint;

        schSCManager = OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);
        schService = OpenService(schSCManager, TEXT("W3SVC"), SERVICE_ALL_ACCESS);
        if (schService == NULL)
                return FALSE;

        if (! QueryServiceStatus(schService, &ssStatus) )
                goto StopWWWebErr;

        if ( !ControlService(schService, SERVICE_CONTROL_STOP, &ssStatus) )
                goto StopWWWebErr;
        //start Service pending, Check the status until the service is running.
        if (! QueryServiceStatus(schService, &ssStatus) )
                goto StopWWWebErr;
        while( ssStatus.dwCurrentState == SERVICE_RUNNING)
        {

                dwOldCheckPoint = ssStatus.dwCheckPoint;                        //Save the current
checkpoint.
                Sleep(ssStatus.dwWaitHint);
        //Wait for the specified interval.
                if ( !QueryServiceStatus(schService, &ssStatus) )    //Check the status again.
                        break;
```

```
                        if (dwOldCheckPoint >= ssStatus.dwCheckPoint)           //Break if the checkpoint has not
been incremented.
                                break;
        }

        if (ssStatus.dwCurrentState == SERVICE_RUNNING)
                goto StopWWWebErr;

        CloseServiceHandle(schService);
        return TRUE;

StopWWWebErr:
        CloseServiceHandle(schService);
        return FALSE;
}

static void UpdateDialog(HWND hDlg)
{
        MSG msg;

        UpdateWindow(hDlg);
        while( PeekMessage(&msg, hDlg, 0, 0, PM_REMOVE) )
        {
                TranslateMessage(&msg);
                DispatchMessage(&msg);
        }
        Sleep(250);
        return;
}
```

install_com.cpp

```
/*      FILE:                   INSTALL_COM.CPP
 *                                      Microsoft TPC-C Kit Ver. 4.20.000
 *                                      Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                                      not audited
 *
 *      PURPOSE:  installation code for COM application for TPC-C Web Kit
 *      Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                      4.20.000 - first version
 */

#define _WIN32_WINNT 0x0500

#include <comdef.h>
#include <comadmin.h>
#include <stdio.h>
#include <tchar.h>

extern "C"
{
        BOOL install_com(char *szDllPath);
}

BOOL install_com(char *szDllPath)
{
        ICOMAdminCatalog*   pCOMAdminCat = NULL;
        ICatalogCollection* pCatalogCollectionApp    = NULL;
        ICatalogCollection* pCatalogCollectionCo     = NULL;
        ICatalogCollection* pCatalogCollectionItf    = NULL;
        ICatalogCollection* pCatalogCollectionMethod = NULL;

        ICatalogObject*             pCatalogObjectApp    = NULL;
        ICatalogObject*             pCatalogObjectCo     = NULL;
        ICatalogObject*             pCatalogObjectItf    = NULL;
        ICatalogObject*             pCatalogObjectMethod = NULL;

        _bstr_t                     bstrTemp, bstrTemp2, bstrTemp3, bstrTemp4;
        _bstr_t                     bstrDllPath = szDllPath;
        _variant_t                  vTmp, vKey;
        long                        lActProp, lCount, lCountCo, lCountItf, lCountMethod;
        bool                        bTmp;


        CoInitializeEx(NULL, COINIT_MULTITHREADED);

        HRESULT hr = CoCreateInstance(CLSID_COMAdminCatalog,
                                                                                                NULL,
CLSCTX_INPROC_SERVER,

IID_ICOMAdminCatalog,
                                                                                                (void**)
&pCOMAdminCat);

        if (!SUCCEEDED(hr)) goto Error;

        bstrTemp = "Applications";

        // Attempt to connect to "Applications" in the Catalog
        hr = pCOMAdminCat->GetCollection(bstrTemp,

(IDispatch**) &pCatalogCollectionApp);
        if (!SUCCEEDED(hr)) goto Error;

        // Attempt to load the "Applications" collection
        hr = pCatalogCollectionApp->Populate();
        if (!SUCCEEDED(hr)) goto Error;

        hr = pCatalogCollectionApp->get_Count(&lCount);
        if (!SUCCEEDED(hr)) goto Error;

        // iterate through applications to delete existing"TPC-C" application (if any)
        while (lCount > 0)
        {
                hr = pCatalogCollectionApp->get_Item(lCount- 1, (IDispatch**) &pCatalogObjectApp);
```

```
		if (!SUCCEEDED(hr)) goto Error;

		hr = pCatalogObjectApp->get_Name(&vTmp);
		if (!SUCCEEDED(hr)) goto Error;

		if (wcscmp(vTmp.bstrVal, L"TPC-C"))
		{
			lCount--;
			continue;
		}
		else
		{
			hr = pCatalogCollectionApp->Remove(lCount- 1);
			if (!SUCCEEDED(hr)) goto Error;
			break;
		}
}

hr = pCatalogCollectionApp->SaveChanges(&lActProp);
if (!SUCCEEDED(hr)) goto Error;

// add the new application
hr = pCatalogCollectionApp->Add((IDispatch**)&pCatalogObjectApp);
if (!SUCCEEDED(hr)) goto Error;

// set properties
bstrTemp = "Name";
vTmp = "TPC-C";
hr = pCatalogObjectApp->put_Value(bstrTemp, vTmp);
if (!SUCCEEDED(hr)) goto Error;

// set as a library (in process) application
bstrTemp = "Activation";
lActProp = COMAdminActivationInproc;
vTmp = lActProp;
hr = pCatalogObjectApp->put_Value(bstrTemp, vTmp);
if (!SUCCEEDED(hr)) goto Error;

// set security level to process
bstrTemp = "AccessChecksLevel";
lActProp = COMAdminAccessChecksApplicationLevel;
vTmp = lActProp;
hr = pCatalogObjectApp->put_Value(bstrTemp, vTmp);
if (!SUCCEEDED(hr)) goto Error;

// save key to get the Components collection later
hr = pCatalogObjectApp->get_Key(&vKey);
if (!SUCCEEDED(hr)) goto Error;

// save changes (app creation) so component installation will work
hr = pCatalogCollectionApp->SaveChanges(&lActProp);
if (!SUCCEEDED(hr)) goto Error;

pCatalogObjectApp->Release();
pCatalogObjectApp = NULL;
```

```
bstrTemp  = "TPC-C";                                    // app name
bstrTemp2 =bstrDllPath + "tpcc_com_all.dll";            // DLL
bstrTemp3 ="";                                                  // type library (TLB)
bstrTemp4 =bstrDllPath + "tpcc_com_ps.dll";             // proxy/stub dll

hr = pCOMAdminCat->InstallComponent(bstrTemp,

bstrTemp2,

bstrTemp3,

bstrTemp4);
if (!SUCCEEDED(hr)) goto Error;


bstrTemp = "Components";
hr = pCatalogCollectionApp->GetCollection(bstrTemp,vKey, (IDispatch**) &pCatalogCollectionCo);
if (!SUCCEEDED(hr)) goto Error;

hr = pCatalogCollectionCo->Populate();
if (!SUCCEEDED(hr)) goto Error;

hr = pCatalogCollectionCo->get_Count(&lCountCo);
if (!SUCCEEDED(hr)) goto Error;

// iterate through components in application and set the properties
while (lCountCo > 0)
{
		hr = pCatalogCollectionCo->get_Item(lCountCo- 1, (IDispatch**) &pCatalogObjectCo);
		if (!SUCCEEDED(hr)) goto Error;

		// used for debugging (view the name)
		hr = pCatalogObjectCo->get_Name(&vTmp);
		if (!SUCCEEDED(hr)) goto Error;

		bstrTemp = "ConstructionEnabled";
		bTmp = TRUE;
		vTmp = bTmp;
		hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
		if (!SUCCEEDED(hr)) goto Error;

		bstrTemp = "ConstructorString";
		bstrTemp2 = "dummy string (do not remove)";
		vTmp = bstrTemp2;
		hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
		if (!SUCCEEDED(hr)) goto Error;

		bstrTemp = "JustInTimeActivation";
		bTmp = TRUE;
		vTmp = bTmp;
		hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
		if (!SUCCEEDED(hr)) goto Error;

		bstrTemp = "MaxPoolSize";
		vTmp.Clear();		// clear variant so it isn't stored as a bool (_variant_t feature)
		vTmp = (long)30;
```

```
                hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
                if (!SUCCEEDED(hr)) goto Error;

                bstrTemp = "ObjectPoolingEnabled";
                bTmp = TRUE;
                vTmp = bTmp;
                hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
                if (!SUCCEEDED(hr)) goto Error;



                // save key to get the InterfacesForComponent collection
                hr = pCatalogObjectCo->get_Key(&vKey);
                if (!SUCCEEDED(hr)) goto Error;

                bstrTemp = "InterfacesForComponent";
                hr = pCatalogCollectionCo->GetCollection(bstrTemp,vKey, (IDispatch**)
&pCatalogCollectionItf);
                if (!SUCCEEDED(hr)) goto Error;

                hr = pCatalogCollectionItf->Populate();
                if (!SUCCEEDED(hr)) goto Error;

                hr = pCatalogCollectionItf->get_Count(&lCountItf);
                if (!SUCCEEDED(hr)) goto Error;

                // iterate through interfaces in component
                while (lCountItf > 0)
                {
                        hr = pCatalogCollectionItf->get_Item(lCountItf- 1, (IDispatch**)
&pCatalogObjectItf);
                        if (!SUCCEEDED(hr)) goto Error;

                        // save key to get the MethodsForInterface collection
                        hr = pCatalogObjectItf->get_Key(&vKey);
                        if (!SUCCEEDED(hr)) goto Error;

                        bstrTemp = "MethodsForInterface";
                        hr = pCatalogCollectionItf->GetCollection(bstrTemp,vKey, (IDispatch**)
&pCatalogCollectionMethod);
                        if (!SUCCEEDED(hr)) goto Error;

                        hr = pCatalogCollectionMethod->Populate();
                        if (!SUCCEEDED(hr)) goto Error;

                        hr = pCatalogCollectionMethod->get_Count(&lCountMethod);
                        if (!SUCCEEDED(hr)) goto Error;

                        // iterate through methods of interface
                        while (lCountMethod > 0)
                        {
                                hr = pCatalogCollectionMethod->get_Item(lCountMethod- 1,
(IDispatch**) &pCatalogObjectMethod);
                                if (!SUCCEEDED(hr)) goto Error;

                                bstrTemp = "AutoComplete";
                                bTmp = TRUE;
                                vTmp = bTmp;
                                hr = pCatalogObjectMethod->put_Value(bstrTemp, vTmp);
                                if (!SUCCEEDED(hr)) goto Error;

                                pCatalogObjectMethod->Release();
                                pCatalogObjectMethod = NULL;

                                lCountMethod--;
                        }

                        // save changes
                        hr = pCatalogCollectionMethod->SaveChanges(&lActProp);
                        if (!SUCCEEDED(hr)) goto Error;

                        pCatalogObjectItf->Release();
                        pCatalogObjectItf = NULL;

                        lCountItf--;

                }


                pCatalogObjectCo->Release();
                pCatalogObjectCo = NULL;

                lCountCo--;
        }


        // save changes
        hr = pCatalogCollectionCo->SaveChanges(&lActProp);
        if (!SUCCEEDED(hr)) goto Error;

        pCatalogCollectionApp->Release();
        pCatalogCollectionApp = NULL;

        pCatalogCollectionCo->Release();
        pCatalogCollectionCo = NULL;

        pCatalogCollectionItf->Release();
        pCatalogCollectionItf= NULL;

        pCatalogCollectionMethod->Release();
        pCatalogCollectionMethod = NULL;


Error:
        CoUninitialize();

        if (!SUCCEEDED(hr))
        {
                LPTSTR lpBuf;
                DWORD dwRes = FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER|
FORMAT_MESSAGE_FROM_SYSTEM,
                                                                                NULL,
```

```
MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),

(LPTSTR) &lpBuf,


//                      _tprintf(__T("Error adding components.  HRESULT: 0x%x\n%s"), hr, lpBuf);
                        return TRUE;
                }
                else
                        return FALSE;
}
```

install_resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//
#define IDD_DIALOG1                     101
#define IDI_ICON1                       102
#define IDR_TPCCDLL                     103
#define IDD_DIALOG2                     105
#define IDI_ICON2                       106
#define IDR_DELIVERY                    107
#define IDD_DIALOG3                     108
#define IDR_LICENSE1                    112
#define IDD_DIALOG4                     113
#define IDR_TPCCOBJ1                    117
#define IDR_TPCCSTUB1                   118
#define IDR_DBLIB_DLL                   122
#define IDR_ODBC_DLL                    123
#define IDR_TUXEDO_APP                  124
#define IDR_TUXEDO_DLL                  125
#define IDR_COM_DLL                     126
#define IDR_COMPS_DLL                   127
#define IDR_COMALL_DLL                  128
#define BN_LOG                          1001
#define ED_KEEP                         1002
#define ED_THREADS                      1003
#define ED_THREADS2                     1004
#define IDC_PATH                        1007
#define IDC_VERSION                     1009
#define IDC_RESULTS                     1010
#define IDC_PROGRESS1                   1011
#define IDC_STATUS                      1012
#define IDC_BUTTON1                     1013
#define ED_MAXCONNECTION                1014
#define ED_IIS_MAX_THEAD_POOL_LIMIT     1015
#define ED_MAXDELIVERIES                1016
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT           1018
#define ED_IIS_LISTEN_BACKLOG           1019
```

```
hr,


0,
NULL);
```

```
#define IDC_DBLIB                       1021
#define IDC_LICENSE                     1022
#define IDC_ODBC                        1022
#define IDC_CONNECT_POOL                1023
#define ED_DB_SERVER                    1023
#define ED_USER_CONNECT_DELAY_TIME      1024
#define ED_DB_USER_ID                   1024
#define IDC_MTS                         1025
#define IDC_TM_MTS                      1025
#define IDC_TM_TUXEDO                   1026
#define IDC_TM_NONE                     1027
#define ED_DB_PASSWORD                  1028
#define ED_DB_NAME                      1029
#define IDC_TM_ENCINA                   1030

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        129
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1024
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

isapi_dll.dsp

```
# Microsoft Developer Studio Project File - Name="isapi_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=isapi_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "isapi_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "isapi_dll.mak" CFG="isapi_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "isapi_dll - Win32 Release" (based on\
 "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 Debug" (based on\
 "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 IceCAP" (based on\
 "Win32 (x86) Dynamic-Link Library")
!MESSAGE
```

```
# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "isapi_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "NDEBUG" /D "WIN32" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\common\txnlog\lib\release\rtetime.lib..\common\txnlog\lib\release\spinlock.lib
..\common\txnlog\lib\release\error.lib..\common\txnlog\lib\release\txnlog.libwsock32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /machine:I386 /out:".\bin/tpcc.dll"

!ELSEIF  "$(CFG)" == "isapi_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MDd /W3 /GX /Zi /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ..\common\txnlog\lib\debug\rtetime.lib..\common\txnlog\lib\debug\spinlock.lib
..\common\txnlog\lib\debug\error.lib..\common\txnlog\lib\debug\txnlog.libwsock32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none

!ELSEIF  "$(CFG)" == "isapi_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "isapi_dl"
# PROP BASE Intermediate_Dir "isapi_dl"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /GX /Zi /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR /YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /GX /Zi /O2 /D "NDEBUG" /D "ICECAP" /D "WIN32" /D "_WINDOWS" /FR /YX /FD
/Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc.dll"
/pdbtype:sept
# SUBTRACT BASE LINK32 /profile /pdb:none
# ADD LINK32 icap.lib ..\common\txnlog\lib\release\rtetime.lib..\common\txnlog\lib\release\spinlock.lib
..\common\txnlog\lib\release\error.lib..\common\txnlog\lib\release\txnlog.libwsock32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /map

!ENDIF

# Begin Target

# Name "isapi_dll - Win32 Release"
# Name "isapi_dll - Win32 Debug"
# Name "isapi_dll - Win32 IceCAP"
# Begin Group "Source"
```

```
# PROP Default_Filter "*.cpp, *.def, *.rc"
# Begin Source File

SOURCE=.\src\tpcc.cpp
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.def
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.rc
# End Source File
# End Group
# Begin Group "Header Files"

# PROP Default_Filter "*.h, *.hpp"
# Begin Source File

SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=..\common\src\ReadRegistry.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.h
# End Source File
# Begin Source File

SOURCE=..\db_dblib_dll\src\tpcc_dblib.h
# End Source File
# Begin Source File

SOURCE=..\db_odbc_dll\src\tpcc_odbc.h
# End Source File
# Begin Source File

SOURCE=..\tm_tuxedo_dll\src\tpcc_tux.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project
```

isapi_dll_resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc.rc
//
#define IDD_DIALOG1                     101

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        102
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1000
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

readregistry.cpp

```
/*              FILE:                   READREGISTRY.CPP
 *                                              Microsoft TPC-C Kit Ver. 4.20.000
 *                                              Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                              not yet audited
 *
 *              PURPOSE:  Implementation for TPC-C Tuxedo class.
 *              Contact:    Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                          4.20.000 - first version
 */


/* FUNCTION: ReadTPCCRegistrySettings
 *
 * PURPOSE:             This function reads the NT registry for startup parameters. There parameters are
 *                              under the TPCC key.
 *
 * RETURNS              FALSE = no errors
 *                              TRUE  = error reading registry
 */
BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg )
{
            HKEY        hKey;
            DWORD       size;
            DWORD       type;
            DWORD       dwTmp;
            char        szTmp[256];
```

```
	if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,"SOFTWARE\\Microsoft\\TPCC",0, KEY_READ, &hKey)
!= ERROR_SUCCESS )
		return TRUE;

	// determine database protocol to use; may be either ODBC or DBLIB
	pReg->eDB_Protocol = Unspecified;
	size = sizeof(szTmp);
	if ( RegQueryValueEx(hKey, "DB_Protocol", 0, &type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS )
	{
		if ( !stricmp(szTmp, szDBNames[ODBC]) )
			pReg->eDB_Protocol = ODBC;
		else if ( !stricmp(szTmp, szDBNames[DBLIB]) )
			pReg->eDB_Protocol = DBLIB;
	}

	pReg->eTxnMon = None;
	// determine txn monitor to use; may be either TUXEDO, or blank
	size = sizeof(szTmp);
	if ( RegQueryValueEx(hKey, "TxnMonitor", 0, &type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS )
	{
		if ( !stricmp(szTmp, szTxnMonNames[TUXEDO]) )
			pReg->eTxnMon = TUXEDO;
		else if ( !stricmp(szTmp, szTxnMonNames[ENCINA]) )
			pReg->eTxnMon = ENCINA;
		else if ( !stricmp(szTmp, szTxnMonNames[COM]) )
			pReg->eTxnMon = COM;
	}

	pReg->bCOM_SinglePool = FALSE;
	size = sizeof(szTmp);
	if ( RegQueryValueEx(hKey, "COM_SinglePool", 0, &type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS
)
	{
		if ( !stricmp(szTmp, "YES") )
			pReg->bCOM_SinglePool = TRUE;
	}

	pReg->dwMaxConnections = 0;
	size = sizeof(dwTmp);
	if ( ( RegQueryValueEx(hKey, "MaxConnections", 0, &type, (LPBYTE)&dwTmp, &size) ==
ERROR_SUCCESS )
		&& (type == REG_DWORD) )
		pReg->dwMaxConnections = dwTmp;

	pReg->dwMaxPendingDeliveries = 0;
	size = sizeof(dwTmp);
	if ( ( RegQueryValueEx(hKey, "MaxPendingDeliveries", 0, &type, (LPBYTE)&dwTmp, &size) ==
ERROR_SUCCESS )
		&& (type == REG_DWORD) )
		pReg->dwMaxPendingDeliveries = dwTmp;

	pReg->dwNumberOfDeliveryThreads = 0;
	size = sizeof(dwTmp);
	if ( ( RegQueryValueEx(hKey, "NumberOfDeliveryThreads", 0, &type, (LPBYTE)&dwTmp, &size) ==
ERROR_SUCCESS )
		&& (type == REG_DWORD) )
		pReg->dwNumberOfDeliveryThreads = dwTmp;

	size = sizeof( pReg->szPath );
	if ( RegQueryValueEx(hKey, "Path", 0, &type, (BYTE *)&pReg->szPath, &size) != ERROR_SUCCESS )
		pReg->szPath[0] = 0;

	size = sizeof( pReg->szDbServer );
	if ( RegQueryValueEx(hKey, "DbServer", 0, &type, (BYTE *)&pReg->szDbServer, &size) !=
ERROR_SUCCESS )
		pReg->szDbServer[0] = 0;

	size = sizeof( pReg->szDbName );
	if ( RegQueryValueEx(hKey, "DbName", 0, &type, (BYTE *)&pReg->szDbName, &size) !=
ERROR_SUCCESS )
		pReg->szDbName[0] = 0;

	size = sizeof( pReg->szDbUser );
	if ( RegQueryValueEx(hKey, "DbUser", 0, &type, (BYTE *)&pReg->szDbUser, &size) !=
ERROR_SUCCESS )
		pReg->szDbUser[0] = 0;

	size = sizeof( pReg->szDbPassword );
	if ( RegQueryValueEx(hKey, "DbPassword", 0, &type, (BYTE *)&pReg->szDbPassword, &size) !=
ERROR_SUCCESS )
		pReg->szDbPassword[0] = 0;

	RegCloseKey(hKey);

	return FALSE;
}
```

readregistry.h

```
/*	FILE:		ReadRegistry.h
 *				Microsoft TPC-C Kit Ver. 4.20.000
 *				Copyright Microsoft, 1999
 *	All Rights Reserved
 *
 *				not audited
 *
 *	PURPOSE:  Header for registry related code.
 *
 *  Change history:
 *		4.20.000 - first version
 */

enum DBPROTOCOL { Unspecified, ODBC, DBLIB };
const char *szDBNames[] = { "Unspecified", "ODBC", "DBLIB" };

enum TXNMON { None, TUXEDO, ENCINA, COM };
const char *szTxnMonNames[] = { "NONE", "TUXEDO", "ENCINA", "COM" };

//This structure defines the data necessary to keep distinct for each terminal or client connection.
typedef struct _TPCCREGISTRYDATA
```

```
{
        enum DBPROTOCOL eDB_Protocol;
        enum TXNMON eTxnMon;
        BOOL bCOM_SinglePool;
        DWORD dwMaxConnections;
        DWORD dwMaxPendingDeliveries;
        DWORD dwNumberOfDeliveryThreads;
        char szPath[128];
        char szDbServer[32];
        char szDbName[32];
        char szDbUser[32];
        char szDbPassword[32];
} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );
```

rtetime.h

```
/* FILE: rtetime.h : header file
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Authors: Charles Levine, Philip Durr
 *                                      Microsoft Corp.
 */

#define     MAX_JULIAN_TIME             0x7FFFFFFFFFFFFFFF
#define JULIAN_TIME  __int64
#define     TC_TIME             DWORD
extern "C"
{
BOOL                InitJulianTime(LPSYSTEMTIME lpInitTime);
JULIAN_TIME         GetJulianTime(void);
DWORD               MyTickCount(void);
void                GetJulianAndTC(JULIAN_TIME *pJulian, DWORD *pTC);
JULIAN_TIME         ConvertTo64BitTime(int iYear, int iMonth, int iDay, int iHour, int iMinute, int iSecond);
JULIAN_TIME         Get64BitTime(LPSYSTEMTIME lpInitTime);
int                         JulianDay( int yr, int mm, int dd );
void                JulianToTime(JULIAN_TIME julianTS, int* yr, int* mm, int* dd, int *hh, int *mi, int *ss );
void                JulianToCalendar( int day, int* yr, int* mm, int* dd );
}
```

spinlock.h

```
/*          FILE: SPINLOCK.H
 *
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Authors: Mike Parkes, Charles Levine, Philip Durr
 *                                      Microsoft Corp.
 */

#ifndef _INC_Spinlock

        const LONG LockClosed = 1;
        const LONG LockOpen             = 0;

        /************************************************************
         *
         *  Spinlock and Semaphore locking.
         *
         *  This class provides a very conservative locking scheme.
         *  The assumption behind the code is that locks will be
         *  held for a very short time.  When a lock is taken a memory
         *  location is exchanged.  All other threads that want this
         *  lock wait by spinning and sometimes sleeping on a semaphore
         *  until it becomes free again.  The only other choice is not
         *  to wait at all and move on to do something else.  This
         *  module should normally be used in conjunction with cache
         *  aligned memory in minimize cache line misses.
         *
         ************************************************************/

        class Spinlock
        {
                // Private data.
                HANDLE          Semaphore;
                volatile LONG    m_Spinlock;
                volatile LONG    Waiting;

                #ifdef _DEBUG
                        // Counters for debugging builds.
                        volatile LONG       TotalLocks;
                        volatile LONG       TotalSleeps;
                        volatile LONG       TotalSpins;
                        volatile LONG       TotalWaits;
                #endif

                public:
                        // Public functions.

                        Spinlock( void );

                        inline BOOL ClaimLock( BOOL Wait = TRUE );
                        inline void ReleaseLock( void );
                        ~Spinlock( void );
                        // Disabled operations.
                        Spinlock( const Spinlock & Copy );
                        void operator=( const Spinlock & Copy );

                private:
                        // Private functions.
                        inline BOOL ClaimSpinlock( volatile LONG *sl );
                        void WaitForLock( void );
                        void WakeAllSleepers( void );
        };

        /************************************************************
```

```
 *
 *   A guaranteed atomic exchange.
 *
 *   An attempt is made to claim the Spinlock.  This action is
 *   guaranteed to be atomic.
 *
 ************************************************************/

inline BOOL Spinlock::ClaimSpinlock(volatile LONG *Spinlock )
{
        #ifdef _DEBUG
                InterlockedIncrement( (LPLONG) & TotalLocks );
        #endif
        return ( ((*Spinlock) == LockOpen) && (InterlockedExchange( (LPLONG)Spinlock, LockClosed
) == LockOpen) );
}

/************************************************
 *
 *   Claim the Spinlock.
 *
 *   Claim the lock if available else wait or exit.
 *
 ************************************************/

inline BOOL Spinlock::ClaimLock(BOOL Wait )
{
        if ( ! ClaimSpinlock( (volatile LONG*) & m_Spinlock ) )
        {
                if ( Wait )
                        WaitForLock();
                return Wait;
        }
        return TRUE;
}

/******************************************************
 *
 *   Release the Spinlock.
 *
 *   Release the lock and if needed wakeup any sleepers.
 *
 ******************************************************/

inline void Spinlock::ReleaseLock( void )
{
        m_Spinlock = LockOpen;
        if ( Waiting > 0 )
                WakeAllSleepers();
}

#define _INC_Spinlock

#endif
```

tm_com_dll.dsp

```
# Microsoft Developer Studio Project File - Name="tm_com_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=tm_com_dll - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tm_com_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tm_com_dll.mak" CFG="tm_com_dll - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tm_com_dll - Win32 Release" (based on\
 "Win32 (x86) Dynamic-Link Library")
!MESSAGE "tm_com_dll - Win32 Debug" (based on\
 "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tm_com_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
```

```
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386 /out:".\bin/tpcc_com.dll"

!ELSEIF  "$(CFG)" == "tm_com_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_com.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "tm_com_dll - Win32 Release"
# Name "tm_com_dll - Win32 Debug"
# Begin Source File

SOURCE=.\src\tpcc_com.cpp
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com.h
# End Source File
# End Target
# End Project
```

tpcc.cpp

```
/*          FILE:                   TPCC.C
 *                                          Microsoft TPC-C Kit Ver. 4.20.000
 *                                          Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                          Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 *          PURPOSE:  Main module for TPCC.DLL which is an ISAPI service dll.
 *          Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                    4.20.000 - reworked error handling; added options for COM and Encina txn monitors
 */

#include <windows.h>
#include <process.h>
#include <tchar.h>
#include <stdio.h>
#include <stdarg.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <io.h>
#include <assert.h>

#include <sqltypes.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

#include "..\..\common\src\trans.h"             //tpckit transaction header contains definations of structures specific to
TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\ReadRegistry.h"

#include "..\..\common\txnlog\include\rtetime.h"
#include "..\..\common\txnlog\include\spinlock.h"
#include "..\..\common\txnlog\include\txnlog.h"

// Database layer includes
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"          // DBLIB implementation ofTPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"            // ODBC implementation ofTPC-C txns

// Txn monitor layer includes
#include "..\..\tm_com_dll\src\tpcc_com.h"                      // COM Services implementation onTPC-C
txns
#include "..\..\tm_tuxedo_dll\src\tpcc_tux.h"           // interface to Tuxedo libraries
#include "..\..\tm_encina_dll\src\tpcc_enc.h"           // interface to Encina libraries

#include "httpext.h"                                    //ISAPI DLL information header
#include "tpcc.h"                                               //this dlls specific structure, value e.t. header.
```

```
#define       LEN_ERR_STRING      256

// defines for Make<Txn>Form calls to distinguish input and output flavors
#define       OUTPUT_FORM         0
#define INPUT_FORM   1

char                        szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

//Terminal client id structure
TERM        Term = { 0, 0, 0, NULL };

// The WEBCLIENT_VERSION string specifies the version level of this web client interface.
// The RTE must be synchronized with the interface level onlogin, otherwise the login
// will fail.  This is a sanity check to catch problems resulting from mismatched versions
// of the RTE and web client.
#define WEBCLIENT_VERSION "410"

static        CRITICAL_SECTION              TermCriticalSection;

static  HINSTANCE hLibInstanceTm = NULL;
static  HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB   *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC                *pCTPCC_ODBC_new;
TYPE_CTPCC_TUXEDO              *pCTPCC_TUXEDO_new;
TYPE_CTPCC_ENCINA   *pCTPCC_ENCINA_new;
TYPE_CTPCC_ENCINA   *pCTPCC_ENCINA_post_init;
TYPE_CTPCC_COM                 *pCTPCC_COM_new;

// For deferred Delivery txns:

CTxnLog                                         *txnDelilog = NULL;                         //used
to log delivery transaction information

HANDLE                                          hWorkerSemaphore      = INVALID_HANDLE_VALUE;
HANDLE                                          hDoneEvent                  =
INVALID_HANDLE_VALUE;
HANDLE                                          *pDeliHandles               = NULL;

// configuration settings from registry
TPCCREGISTRYDATA            Reg;


DWORD                                           dwNumDeliveryThreads = 4;
CRITICAL_SECTION            DelBuffCriticalSection;         //critical section for delivery transactions cache
DELIVERY_TRANSACTION        *pDelBuff                   = NULL;
DWORD                                           dwDelBuffSize               = 100;
// size of circular buffer for deliverytxns
DWORD                                           dwDelBuffFreeCount;
// number of buffers free
DWORD                                           dwDelBuffBusyIndex    = 0;                  // index
position of entry waiting to be delivered
DWORD                                           dwDelBuffFreeIndex    = 0;                  // index
position of unused entry
```

```
#include "..\..\common\src\ReadRegistry.cpp"

/* FUNCTION: DllMain
 *
 * PURPOSE:          This function is the entry point for the DLL.  This implementation is based on the
 *                   fact that DLL_PROCESS_ATTACH is only called from the inet service once.
 *
 * ARGUMENTS:        HANDLE    hModule                         module handle
 *                             DWORD     ul_reason_for_call    reason for call
 *                             LPVOID    lpReserved                  reserved for future
use
 *
 * RETURNS:                    BOOL      FALSE                         errors occured in
initialization
 *                                                 TRUE
DLL successfully initialized
 */

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
        DWORD  i;
        char szEvent[LEN_ERR_STRING] = "\0";
        char szLogFile[128];
        char szDllName[128];

        try
        {
                switch( ul_reason_for_call )
                {
                        case DLL_PROCESS_ATTACH:
                                {
                                        DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;
                                        GetComputerName(szMyComputerName, &dwSize);
                                        szMyComputerName[dwSize] = 0;
                                }

                                DisableThreadLibraryCalls((HMODULE)hModule);
                                InitializeCriticalSection(&TermCriticalSection);

                                if ( ReadTPCCRegistrySettings( &Reg ) )
                                        throw new CWEBCLNT_ERR(
ERR_MISSING_REGISTRY_ENTRIES );

                                dwDelBuffSize = min( Reg.dwMaxPendingDeliveries, 10000 );  // min
with 10000 as a sanity constraint
                                dwNumDeliveryThreads = min( Reg.dwNumberOfDeliveryThreads, 100
);  // min with 100 as a sanity constraint

                                TermInit();

                                // load DLL for txn monitor
                                if (Reg.eTxnMon == TUXEDO)
                                {
                                        strcpy( szDllName, Reg.szPath );
```

```
                                        strcat( szDllName, "tpcc_tuxedo.dll");
                                        hLibInstanceTm = LoadLibrary( szDllName );
                                        if (hLibInstanceTm == NULL)
                                                throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                                        // get function pointer to wrapper for class constructor
                                        pCTPCC_TUXEDO_new = (TYPE_CTPCC_TUXEDO*)
GetProcAddress(hLibInstanceTm,"CTPCC_TUXEDO_new");
                                        if (pCTPCC_TUXEDO_new == NULL)
                                                throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                }
                                else if (Reg.eTxnMon == ENCINA)
                                {
                                        strcpy( szDllName, Reg.szPath );
                                        strcat( szDllName, "tpcc_encina.dll");
                                        hLibInstanceTm = LoadLibrary( szDllName );
                                        if (hLibInstanceTm == NULL)
                                                throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );
                                        // get function pointer to wrapper for class constructor
                                        pCTPCC_ENCINA_new = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_new");
                                        pCTPCC_ENCINA_post_init =
(TYPE_CTPCC_ENCINA*) GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_post_init");
                                        if (pCTPCC_ENCINA_new == NULL)
                                                throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                }
                                else if (Reg.eTxnMon == COM)
                                {
                                        strcpy( szDllName, Reg.szPath );
                                        strcat( szDllName, "tpcc_com.dll");
                                        hLibInstanceTm = LoadLibrary( szDllName );
                                        if (hLibInstanceTm == NULL)
                                                throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                                        // get function pointer to wrapper for class constructor
                                        pCTPCC_COM_new = (TYPE_CTPCC_COM*)
GetProcAddress(hLibInstanceTm,"CTPCC_COM_new");
                                        if (pCTPCC_COM_new == NULL)
                                                throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                }

                                // load DLL for database connection
                                if ((Reg.eTxnMon == None) || (dwNumDeliveryThreads > 0))
                                {
                                        if (Reg.eDB_Protocol == DBLIB)
                                        {
                                                strcpy( szDllName, Reg.szPath );
                                                strcat( szDllName, "tpcc_dblib.dll");
                                                hLibInstanceDb = LoadLibrary( szDllName );
                                                if (hLibInstanceDb == NULL)
                                                        throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                                                // get function pointer to wrapper for class
constructor
                                                pCTPCC_DBLIB_new =
(TYPE_CTPCC_DBLIB*) GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
                                                if (pCTPCC_DBLIB_new == NULL)
                                                        throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                        }
                                        else if (Reg.eDB_Protocol == ODBC)
                                        {
                                                strcpy( szDllName, Reg.szPath );
                                                strcat( szDllName, "tpcc_odbc.dll");
                                                hLibInstanceDb = LoadLibrary( szDllName );
                                                if (hLibInstanceDb == NULL)
                                                        throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                                                // get function pointer to wrapper for class
constructor
                                                pCTPCC_ODBC_new =
(TYPE_CTPCC_ODBC*) GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                                if (pCTPCC_ODBC_new == NULL)
                                                        throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                        }
                                }

                                if (dwNumDeliveryThreads)
                                {
                                        // for deferred delivery txns:
                                        hDoneEvent = CreateEvent( NULL, TRUE /* manual reset
*/, FALSE /* initially not signalled */, NULL );
                                        InitializeCriticalSection(&DelBuffCriticalSection);
                                        hWorkerSemaphore = CreateSemaphore( NULL, 0,
dwDelBuffSize, NULL );
                                        dwDelBuffFreeCount = dwDelBuffSize;

                                        InitJulianTime(NULL);

                                        // create unique log file name based on
delilog-yymmdd-hhmm.log
                                        SYSTEMTIME Time;
                                        GetLocalTime( &Time );
                                        wsprintf( szLogFile,
"%sdelivery-%2.2d%2.2d%2.2d-%2.2d%2.2d.log",
                                                        Reg.szPath, Time.wYear % 100,
Time.wMonth, Time.wDay, Time.wHour, Time.wMinute );
                                        txnDelilog = new CTxnLog(szLogFile,
TXN_LOG_WRITE);

                                        //write event into txn log for START
                                        txnDelilog->WriteCtrlRecToLog(TXN_EVENT_START,
szMyComputerName, sizeof(szMyComputerName));
```

```
                                        // allocate structures for delivery buffers and thread mgmt
                                        pDeliHandles = new HANDLE[dwNumDeliveryThreads];
                                        pDelBuff = new
DELIVERY_TRANSACTION[dwDelBuffSize];
                                        // launch DeliveryWorkerThread to perform actual delivery
txns
                                        for(i=0; i<dwNumDeliveryThreads; i++)
                                        {
                                                pDeliHandles[i] = (HANDLE) _beginthread(
DeliveryWorkerThread, 0, NULL );
                                                if (pDeliHandles[i] ==
INVALID_HANDLE_VALUE)
                                                        throw new CWEBCLNT_ERR(
ERR_DELIVERY_THREAD_FAILED);
                                        }
                                }

                                break;

                        case DLL_PROCESS_DETACH:
                                if (dwNumDeliveryThreads)
                                {
                                        if (txnDelilog != NULL)
                                        {
                                                //write event into txn log for STOP
txnDelilog->WriteCtrlRecToLog(TXN_EVENT_STOP,szMyComputerName, sizeof(szMyComputerName));
                                                // This will do a clean shutdown of the delivery
log file
                                                CTxnLog     *txnDelilogLocal = txnDelilog;
                                                txnDelilog = NULL;
                                                delete txnDelilogLocal;
                                        }

                                        delete [] pDeliHandles;
                                        delete [] pDelBuff;

                                        CloseHandle( hWorkerSemaphore );
                                        CloseHandle( hDoneEvent );
                                        DeleteCriticalSection(&DelBuffCriticalSection);
                                }

                                DeleteCriticalSection(&TermCriticalSection);

                                if (hLibInstanceTm != NULL)
                                        FreeLibrary( hLibInstanceTm );
                                hLibInstanceTm = NULL;

                                if (hLibInstanceDb != NULL)
                                        FreeLibrary( hLibInstanceDb );
                                hLibInstanceDb = NULL;

                                Sleep(500);
                                break;
                                default:
                                        /* nothing */;
                        }
                }
                catch (CBaseErr *e)
                {
                        WriteMessageToEventLog( e->ErrorText() );
                        delete e;
                        TerminateExtension(0);
                        return FALSE;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled exception.  DLL could not load."));
                        TerminateExtension(0);
                        return FALSE;
                }

                return TRUE;
}


/* FUNCTION: GetExtensionVersion
 *
 * PURPOSE:             This function is called by the inet service when the DLL is first loaded.
 *
 * ARGUMENTS:           HSE_VERSION_INFO   *pVer         passed in structure in which to place expected version
number.
 *
 * RETURNS:                             TRUE         inet service expected return value.
 */

BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVer)
{
                pVer->dwExtensionVersion = MAKELONG(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);
                lstrcpyn(pVer->lpszExtensionDesc, "TPC-C Server.", HSE_MAX_EXT_DLL_NAME_LEN);

                // TODO: why do we need this here instead of in the DLL attach?
                if (Reg.eTxnMon == ENCINA)
                        pCTPCC_ENCINA_post_init();

                return TRUE;
}

/* FUNCTION: TerminateExtension
 *
 * PURPOSE:             This function is called by the inet service when the DLL is about to be unloaded.
 *                                      Release all resources in anticipation of being unloaded.
 *
 * RETURNS:                             TRUE         inet service expected return value.
 */

BOOL WINAPI TerminateExtension( DWORD dwFlags )
{
                if (pDeliHandles)
```

```
		{
			SetEvent( hDoneEvent );
			for(DWORD i=0; i<dwNumDeliveryThreads;i++)
				WaitForSingleObject( pDeliHandles[i], INFINITE );
		}

		TermDeleteAll();
		return TRUE;
}


/* FUNCTION: HttpExtensionProc
 *
 * PURPOSE:		This function is the main entry point for theTPCC DLL. The internet service
 *			calls this function passing in thehttp string.
 *
 * ARGUMENTS:	EXTENSION_CONTROL_BLOCK	*pECB		structure pointer to passed in internet
 *
 *			service information.
 *
 * RETURNS:			DWORD	HSE_STATUS_SUCCESS
		connection can be dropped if error
 *
HSE_STATUS_SUCCESS_AND_KEEP_CONN	keep connect valid comment sent
 *
 * COMMENTS:		None
 *
 */

DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB)
{
		int				iCmd, FormId, TermId, iSyncId;
		char			szBuffer[4096];

		int				lpbSize;
		static char	szHeader[] = "200 Ok";
		DWORD			dwSize = 6;		// initial value is strlen(szHeader)
		char			szHeader1[4096];


#ifdef ICECAP
		StartCAP();
#endif

		try
		{
			//process http query
			ProcessQueryString(pECB, &iCmd, &FormId, &TermId, &iSyncId);

			if (TermId != 0)
			{
				if ( TermId < 0 || TermId >= Term.iNumEntries||
Term.pClientData[TermId].iNextFree!= -1 )
				{
					// debugging...
					char szTmp[128];
					wsprintf( szTmp, "Invalid term ID; TermId = %d", TermId );
					WriteMessageToEventLog( szTmp );

					throw new CWEBCLNT_ERR( ERR_INVALID_TERMID );
				}

				//must have a valid syncid here since termid is valid
				if (iSyncId != Term.pClientData[TermId].iSyncId)
					throw new CWEBCLNT_ERR(
ERR_INVALID_SYNC_CONNECTION );

				//set use time
				Term.pClientData[TermId].iTickCount= GetTickCount();
			}


			switch(iCmd)
			{
			case 0:
				WelcomeForm(pECB, szBuffer);
				break;
			case 1:
				switch( FormId )
				{
					case WELCOME_FORM:
					case MAIN_MENU_FORM:
						break;
					case NEW_ORDER_FORM:
						ProcessNewOrderForm(pECB, TermId, szBuffer);
						break;
					case PAYMENT_FORM:
						ProcessPaymentForm(pECB, TermId, szBuffer);
						break;
					case DELIVERY_FORM:
						ProcessDeliveryForm(pECB, TermId, szBuffer);
						break;
					case ORDER_STATUS_FORM:
						ProcessOrderStatusForm(pECB, TermId, szBuffer);
						break;
					case STOCK_LEVEL_FORM:
						ProcessStockLevelForm(pECB, TermId, szBuffer);
						break;
				}
				break;
			case 2:
				// new-order selected from menu; display new-order input form
				MakeNewOrderForm(TermId, NULL, INPUT_FORM, szBuffer);
				break;
			case 3:
				// payment selected from menu; display payment input form
				MakePaymentForm(TermId, NULL, INPUT_FORM, szBuffer);
				break;
			case 4:
				// delivery selected from menu; display delivery input form
				MakeDeliveryForm(TermId, NULL, INPUT_FORM, szBuffer);
				break;
```

```
                case 5:
                        // order-status selected from menu; display order-status input form
                        MakeOrderStatusForm(TermId, NULL, INPUT_FORM, szBuffer);
                        break;
                case 6:
                        // stock-level selected from menu; display stock-level input form
                        MakeStockLevelForm(TermId, NULL, INPUT_FORM, szBuffer);
                        break;
                case 7:
                        // ExitCmd
                        TermDelete(TermId);
                        WelcomeForm(pECB, szBuffer);
                        break;
                case 8:
                        SubmitCmd(pECB, szBuffer);
                        break;
                case 9:
                        // menu
                        MakeMainMenuForm(TermId, Term.pClientData[TermId].iSyncId, szBuffer);
                        break;
                case 10:
                        // CMD=Clear
                        // resets all connections; should only be used when no other connections are active
                        TermDeleteAll();
                        TermInit();
                        WelcomeForm(pECB, szBuffer);
                        break;
                case 11:    // CMD=Stats
                        StatsCmd(pECB, szBuffer);
                        break;
                }
        }
        catch (CBaseErr *e)
        {
                ErrorForm( pECB, e->ErrorType(), e->ErrorNum(), TermId, iSyncId, e->ErrorText(), szBuffer );
                delete e;
        }
        catch (...)
        {
                ErrorForm( pECB, ERR_TYPE_WEBDLL, 0, TermId, iSyncId, "Error: Unhandled exception in
Web Client.", szBuffer );
        }

#ifdef ICECAP
        StopCAP();
#endif


        lpbSize = strlen(szBuffer);
        wsprintf(szHeader1,
                        "Content-Type: text/html\r\n"
                        "Content-Length: %d\r\n"
                        "Connection: Keep-Alive\r\n\r\n", lpbSize);
        strcat( szHeader1, szBuffer );
                (*pECB->ServerSupportFunction)(pECB->ConnID,HSE_REQ_SEND_RESPONSE_HEADER, szHeader,
(LPDWORD) &dwSize, (LPDWORD)szHeader1);

                //finish up and keep connection
                pECB->dwHttpStatusCode = 200;
                return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}


void WriteMessageToEventLog(LPTSTR lpszMsg)
{
   TCHAR   szMsg[256];
   HANDLE  hEventSource;
   LPTSTR  lpszStrings[2];

   // Use event logging to log the error.
   //
   hEventSource = RegisterEventSource(NULL, TEXT("TPCC.DLL"));

   _stprintf(szMsg, TEXT("Error in TPCC.DLL: "));
   lpszStrings[0] = szMsg;
   lpszStrings[1] = lpszMsg;

   if (hEventSource != NULL)
                {
      ReportEvent(hEventSource, // handle of event source
         EVENTLOG_ERROR_TYPE,  // event type
         0,                    // event category
         0,                    // event ID
         NULL,                 // current user's SID
         2,                    // strings in  lpszStrings
         0,                    // no bytes of raw data
         (LPCTSTR *)lpszStrings,    // array of error strings
         NULL);                // no raw data

      (VOID) DeregisterEventSource(hEventSource);
   }
}


/* FUNCTION: DeliveryWorkerThread
 *
 * PURPOSE:              This function processes deferred delivery txns.  There are typically several
 *                                 threads running this routine.  The number of threads is determined by an entry
 *                                 read from the registry.  The thread waits for work by waiting on semaphore.
 *                                 When a delivery txn is posted, the semaphore is released.  After processing
 *                                 the delivery txn, information is logged to record the txn status and execution
 *                                 time.
 */

/*static*/ void DeliveryWorkerThread(void *ptr)
{
        CTPCC_BASE                      *pTxn = NULL;

        DELIVERY_TRANSACTION                    delivery;
        PDELIVERY_DATA                                  pDeliveryData;
```

```
	TXN_RECORD_TPCC_DELIV_DEFtxnDeliRec;

	DWORD                                   index;
	HANDLE                                  handles[2];

	SYSTEMTIME                  trans_end;          //delivery transaction finished time
	SYSTEMTIME                  trans_start;   //delivery transaction start time

	assert(txnDelilog != NULL);

	try
	{
		if (Reg.eDB_Protocol == ODBC)
			pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
		else if (Reg.eDB_Protocol == DBLIB)
			pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
		pDeliveryData = pTxn->BuffAddr_Delivery();
	}
	catch (CBaseErr *e)
	{
		char szTmp[1024];
		wsprintf( szTmp, "Error in Delivery Txn thread.  Could not connect to database.  "
				"%s.  Server=%s, User=%s, Password=%s, Database=%s",
				e->ErrorText(), Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
Reg.szDbName );
		WriteMessageToEventLog( szTmp );
		delete e;
		goto ErrorExit;
	}
	catch (...)
	{
		WriteMessageToEventLog(TEXT("Unhandled exception caught in DeliveryWorkerThread."));
		goto ErrorExit;
	}

	while (TRUE)
	{
		try
		{
			//while delivery thread running, i.e. user has not requested termination
			while (TRUE)
			{
				// need to wait for multiple objects: program exit or worker semaphore;
				handles[0] = hDoneEvent;
				handles[1] = hWorkerSemaphore;
				index = WaitForMultipleObjects( 2, &handles[0], FALSE, INFINITE );
				if (index == WAIT_OBJECT_0)
					goto ErrorExit;

				ZeroMemory(&txnDeliRec, sizeof(txnDeliRec));
				txnDeliRec.TxnType = TXN_REC_TYPE_TPCC_DELIV_DEF;

				// make a local copy of current entry from delivery buffer and increment
buffer index
				EnterCriticalSection(&DelBuffCriticalSection);
				delivery = *(pDelBuff+dwDelBuffBusyIndex);
				dwDelBuffFreeCount++;
				dwDelBuffBusyIndex++;
				if (dwDelBuffBusyIndex == dwDelBuffSize)	// wrap-around if at
end of buffer
					dwDelBuffBusyIndex = 0;

				LeaveCriticalSection(&DelBuffCriticalSection);

				pDeliveryData->w_id = delivery.w_id;
				pDeliveryData->o_carrier_id = delivery.o_carrier_id;

				txnDeliRec.w_id = pDeliveryData->w_id;
				txnDeliRec.o_carrier_id = pDeliveryData->o_carrier_id;
				txnDeliRec.TxnStartT0 = Get64BitTime(&delivery.queue);

				GetLocalTime( &trans_start );
				pTxn->Delivery();
				GetLocalTime( &trans_end );

				//log txn
				txnDeliRec.TxnStatus = ERR_SUCCESS;
				for (int i=0; i<10; i++)
					txnDeliRec.o_id[i] = pDeliveryData->o_id[i];
				txnDeliRec.DeltaT4 = (int)(Get64BitTime(&trans_end)-
txnDeliRec.TxnStartT0);

				txnDeliRec.DeltaTxnExec = (int)(Get64BitTime(&trans_end)-
Get64BitTime(&trans_start));

				if (txnDelilog != NULL)
					txnDelilog->WriteToLog(&txnDeliRec);
			}
		}
		catch (CBaseErr *e)
		{
			char szTmp[1024];
			wsprintf( szTmp, "Error in Delivery Txn thread.  %s", e->ErrorText() );
			WriteMessageToEventLog( szTmp );
			delete e;

			// log the error txn
			txnDeliRec.TxnStatus = e->ErrorType();
			if (txnDelilog != NULL)
				txnDelilog->WriteToLog(&txnDeliRec);
		}
		catch (...)
		{
			// unhandled exception; shouldn't happen; not much we can do...
			WriteMessageToEventLog(TEXT("Unhandled exception caught in
DeliveryWorkerThread."));
		}
	}

ErrorExit:
	delete pTxn;
```

```
        _endthread();
}

/* FUNCTION: PostDeliveryInfo
 *
 * PURPOSE:             This function enters the delivery txn into the deferred delivery buffer.
 *
 * RETURNS:                     BOOL    FALSE   delivery information posted successfully
 *                                              TRUE    error cannot post delivery info
 */

BOOL PostDeliveryInfo(short w_id, short o_carrier_id)
{
        BOOL bError;

        EnterCriticalSection(&DelBuffCriticalSection);
        if (dwDelBuffFreeCount > 0)
        {
                bError = FALSE;
                (pDelBuff+dwDelBuffFreeIndex)->w_id                             = w_id;
                (pDelBuff+dwDelBuffFreeIndex)->o_carrier_id     = o_carrier_id;
                GetLocalTime(&(pDelBuff+dwDelBuffFreeIndex)->queue);

                dwDelBuffFreeCount--;
                dwDelBuffFreeIndex++;
                if (dwDelBuffFreeIndex == dwDelBuffSize)
                        dwDelBuffFreeIndex = 0;         // wrap-around if at end of buffer
        }
        else
                // No free buffers.  Return an error, which indicates that the delivery buffer is full.
                // Most likely, the number of delivery worker threads needs to be increased to keep up
                // with the txn rate.
                bError = TRUE;
        LeaveCriticalSection(&DelBuffCriticalSection);

        if (!bError)
                // increment worker semaphore to wake up a worker thread
                ReleaseSemaphore( hWorkerSemaphore, 1, NULL );

        return bError;
}

/* FUNCTION: ProcessQueryString
 *
 * PURPOSE:             This function extracts the relevent information out of the http command passed in from
 *                              the browser.
 *
 * COMMENTS:            If this is the initial connection i.e. client is at welcome screen then
 *                                      there will not be a terminal id or current form id.  If this is the case
 *                                      then the pTermid and pFormid return values are undefined.
 */

void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int *pFormId, int *pTermId, int
*pSyncId)
{
        char *ptr = pECB->lpszQueryString;
        char szBuffer[25];
        int i;

        //allowable client command strings i.e. CMD=command
        static char *szCmds[] =
        {
                "Process", "..NewOrder..", "..Payment..", "..Delivery..", "..Order-Status..", "..Stock-Level..",
                "..Exit..", "Submit", "Menu", "Clear", "Stats", ""
        };

        *pCmd         = 0;                      // default is the login screen
        *pTermId = 0;

        // if no params (i.e., empty query string), then return login screen
        if (strlen(pECB->lpszQueryString) == 0)
                return;

        // parse FORMID, TERMID, and SYNCID
        *pFormId = GetIntKeyValue(&ptr, "FORMID", NO_ERR, NO_ERR);
        *pTermId = GetIntKeyValue(&ptr, "TERMID", NO_ERR, NO_ERR);
        *pSyncId = GetIntKeyValue(&ptr, "SYNCID", NO_ERR, NO_ERR);

        // parse CMD
        GetKeyValue(&ptr, "CMD", szBuffer, sizeof(szBuffer), ERR_COMMAND_UNDEFINED);

        // see which command it matches
        for(i=0; ; i++)
        {
                if (szCmds[i][0] == 0)
                        // no more; no match; return error
                        throw new CWEBCLNT_ERR( ERR_COMMAND_UNDEFINED );
                if ( !strcmp(szCmds[i], szBuffer) )
                {
                        *pCmd = i+1;
                        break;
                }
        }
}

/* FUNCTION: void WelcomeForm
 *
 */

void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
        char szTmp[1024];

        //welcome to tpc-c html form buffer, this is first form client sees.
        strcpy( szBuffer,               "<HTML><HEAD><TITLE>TPC-C Web Client</TITLE></HEAD><BODY>"
                                                                        "<B><BIG>Microsoft TPC-C Web Client (ver
4.20)</BIG></B> <BR> <BR>"
                                                                        "<font face=\"Courier New\"><PRE>"
                                                                        "Compiled: "__DATE__", "__TIME__"
<BR>"
                                                                        "Source:   "__FILE__" ("__TIMESTAMP__")
<BR>"
```

```
                                                                "</PRE></font>"
                                                                "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"1\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"0\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"0\">"
                                                                "<INPUT TYPE=\"hidden\"
NAME=\"VERSION\" VALUE=\"" WEBCLIENT_VERSION "\">"
                                        );

        sprintf( szTmp,         "Configuration Settings:<BR><font face=\"Courier New\" color=\"blue\"><PRE>"
                                                        "Txn Monitor            = <B>%s</B><BR>"
                                                        "Database protocol      = <B>%s</B><BR>"
                                                        "Max Connections        = <B>%d</B><BR>"
                                                        "# of Delivery Threads  = <B>%d</B><BR>"
                                                        "Max Pending Deliveries = <B>%d</B><BR>"
                                        , szTxnMonNames[Reg.eTxnMon], szDBNames[Reg.eDB_Protocol],
                                        Reg.dwMaxConnections, dwNumDeliveryThreads, dwDelBuffSize );
        strcat( szBuffer, szTmp);

        if (Reg.eTxnMon == COM)
        {
                sprintf( szTmp,         "COM Single Pool        = <B>%s</B><BR>",
                                Reg.bCOM_SinglePool ? "YES" : "NO" );
                strcat( szBuffer, szTmp);
        }
        strcat( szBuffer, "</PRE></font>");

        if (Reg.eTxnMon == None)
                // connection options may be specified when not using atxn monitor
                sprintf( szTmp,         "Please enter your database options for this connection:<BR>"
                                                                "<font face=\"Courier New\"
color=\"blue\"><PRE>"
                                                                "DB Server    = <INPUT
NAME=\"db_server\" SIZE=20 VALUE=\"%s\"><BR>"
                                                                "DB User ID   = <INPUT NAME=\"db_user\"
SIZE=20 VALUE=\"%s\"><BR>"
                                                                "DB Password  = <INPUT
NAME=\"db_passwd\" SIZE=20 VALUE=\"%s\"><BR>"
                                                                "DB Name      = <INPUT
NAME=\"db_name\" SIZE=20 VALUE=\"%s\"><BR>"
                                                                "</PRE></font>"
                                        , Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName
);
        else
                // if using a txn monitor, connection options are determined from registry; can't
                // set per user.  show options fyi
                sprintf( szTmp,         "Database options which will be used by the transaction monitor:<BR>"
                                                                "<font face=\"Courier New\"
color=\"blue\"><PRE>"
                                                                "DB Server            = <B>%s</B><BR>"
                                                                "DB User ID           = <B>%s</B><BR>"
                                                                "DB Password          = <B>%s</B><BR>"
                                                                "DB Name              = <B>%s</B><BR>"
                                                                "</PRE></font>"
                                        , Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName
);
        strcat( szBuffer, szTmp);

        sprintf( szTmp,         "Please enter your Warehouse and District for this session:<BR>"
                                                        "<font face=\"Courier New\" color=\"blue\"><PRE>" );
        strcat( szBuffer, szTmp);
        strcat( szBuffer,       "Warehouse ID = <INPUT NAME=\"w_id\" SIZE=4><BR>"
                                                                "District ID  = <INPUT NAME=\"d_id\"
SIZE=2><BR>"
                                                                "</PRE></font><HR>"
                                                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Submit\">"
                                                                "</FORM></BODY></HTML>");
}

/* FUNCTION: SubmitCmd
 *
 * PURPOSE:             This function allocated a new terminal id in the Term structure array.
 *
 */

void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
        int                     iNewTerm;
        char    *ptr = pECB->lpszQueryString;

        char    szVersion[32]           = { 0 };
        char    szServer[32]= { 0 };
        char    szUser[32]              = "sa";
        char    szPassword[32]          = { 0 };
        char    szDatabase[32]          = "tpcc";

        // validate version field; the version field ensures that the RTE is synchronized with the web client
        GetKeyValue(&ptr, "VERSION", szVersion, sizeof(szVersion), ERR_VERSION_MISMATCH);
        if ( strcmp( szVersion, WEBCLIENT_VERSION ) )
                throw new CWEBCLNT_ERR( ERR_VERSION_MISMATCH );

        if (Reg.eTxnMon == None)
        {
                // parse Server name
                GetKeyValue(&ptr, "db_server", szServer, sizeof(szServer), ERR_NO_SERVER_SPECIFIED);
                // parse User name
                GetKeyValue(&ptr, "db_user", szUser, sizeof(szUser), NO_ERR);
                // parse Password
                GetKeyValue(&ptr, "db_passwd", szPassword, sizeof(szPassword), NO_ERR);
                // parse Database name
                GetKeyValue(&ptr, "db_name", szDatabase, sizeof(szDatabase), NO_ERR);
        }
```

```
	// parse warehouse ID
	int w_id = GetIntKeyValue(&ptr, "w_id", ERR_HTML_ILL_FORMED, ERR_W_ID_INVALID);
	if ( w_id < 1 )
		throw new CWEBCLNT_ERR( ERR_W_ID_INVALID );

	// parse district ID
	int d_id = GetIntKeyValue(&ptr, "d_id", ERR_HTML_ILL_FORMED, ERR_D_ID_INVALID);
	if ( d_id < 1 || d_id > 10 )
		throw new CWEBCLNT_ERR( ERR_D_ID_INVALID );


	iNewTerm = TermAdd();

	Term.pClientData[iNewTerm].w_id= w_id;
	Term.pClientData[iNewTerm].d_id= d_id;

	try
	{
		if (Reg.eTxnMon == TUXEDO)
			Term.pClientData[iNewTerm].pTxn= pCTPCC_TUXEDO_new();
		else if (Reg.eTxnMon == ENCINA)
			Term.pClientData[iNewTerm].pTxn= pCTPCC_ENCINA_new();
		else if (Reg.eTxnMon == COM)
			Term.pClientData[iNewTerm].pTxn= pCTPCC_COM_new( Reg.bCOM_SinglePool
);
		else if (Reg.eDB_Protocol == ODBC)
			Term.pClientData[iNewTerm].pTxn= pCTPCC_ODBC_new( szServer, szUser,
szPassword, szMyComputerName, szDatabase );
		else if (Reg.eDB_Protocol == DBLIB)
			Term.pClientData[iNewTerm].pTxn= pCTPCC_DBLIB_new( szServer, szUser,
szPassword, szMyComputerName, szDatabase );
	}
	catch (...)
	{
		TermDelete(iNewTerm);
		throw;			// pass exception upward
	}

	MakeMainMenuForm(iNewTerm, Term.pClientData[iNewTerm].iSyncId, szBuffer);
}

/* FUNCTION: StatsCmd
 *
 * PURPOSE:		This function returns to the browser the total number of active terminal ids.
 *			This routine is for development/debugging purposes.
 *
 */

void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
	int i;
	int		iTotal;

	EnterCriticalSection(&TermCriticalSection);

	iTotal = 0;
	for(i=0; i<Term.iNumEntries; i++)
	{
		if (Term.pClientData[i].iNextFree == -1)
			iTotal++;
	}

	LeaveCriticalSection(&TermCriticalSection);

	wsprintf( szBuffer,
			"<HTML><HEAD><TITLE>TPC-C Web Client Stats</TITLE></HEAD>"
			"<BODY><B><BIG> Total Active Connections: %d
</BIG></B><BR></BODY></HTML>"
			, iTotal );
}

char *CWEBCLNT_ERR::ErrorText()
{
	static SERRORMSG errorMsgs[] =
	{
		{	ERR_COMMAND_UNDEFINED,
"Command undefined."
			},
		{	ERR_D_ID_INVALID,
"Invalid District ID Must be 1 to 10."
},
		{	ERR_DELIVERY_CARRIER_ID_RANGE,
"Delivery Carrier ID out of range must be 1 - 10."	},
		{	ERR_DELIVERY_CARRIER_INVALID,
"Delivery Carrier ID invalid must be numeric 1 - 10."	},
		{	ERR_DELIVERY_MISSING_OCD_KEY,
"Delivery missing Carrier ID key\"OCD*\"."
},
		{	ERR_DELIVERY_THREAD_FAILED,
"Could not start delivery worker thread."
},
		{	ERR_GETPROCADDR_FAILED,
"Could not map proc in DLL.  GetProcAddr error.  DLL="	},
		{	ERR_HTML_ILL_FORMED,
"Required key field is missing fromHTML string."	},
		{	ERR_INVALID_SYNC_CONNECTION,	"Invalid
Terminal Sync ID."
		},
		{	ERR_INVALID_TERMID,
	"Invalid Terminal ID."
				},
		{	ERR_LOADDLL_FAILED,
	"Load of DLL failed.  DLL="
				},
		{	ERR_MAX_CONNECTIONS_EXCEEDED,	"No
connections available.  Max Connections is probably too low."	},
		{	ERR_MISSING_REGISTRY_ENTRIES,
"Required registry entries are missing.  Rerun INSTALL to correct."	},
		{	ERR_NEWORDER_CUSTOMER_INVALID,	"New
Order customer id invalid data type, range = 1 to 3000."	},
```

```
                {       ERR_NEWORDER_CUSTOMER_KEY,
"New Order missing Customer key \"CID*\"."
},
                {       ERR_NEWORDER_DISTRICT_INVALID,                  "New
Order District ID Invalid range 1 - 10."                                },
                {       ERR_NEWORDER_FORM_MISSING_DID,                  "New
Order missing District key \"DID*\"."                                   },
                {       ERR_NEWORDER_ITEMID_INVALID,                    "New
Order Item Id is wrong data type, must be numeric."                             },
                {       ERR_NEWORDER_ITEMID_RANGE,
"New Order Item Id is out of range. Range = 1 to 999999."                       },
                {       ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,              "New
Order Item_Id field entered without a corresponding Supp_W."    },
                {       ERR_NEWORDER_MISSING_IID_KEY,                   "New
Order missing Item Id key \"IID*\"."                                    },
                {       ERR_NEWORDER_MISSING_QTY_KEY,                   "New
Order Missing Qty key \"Qty##*\"."
},
                {       ERR_NEWORDER_MISSING_SUPPW_KEY,                 "New
Order missing Supp_W key \"SP##*\"."
},
                {       ERR_NEWORDER_NOITEMS_ENTERED,                   "New
Order No order lines entered."
        },
                {       ERR_NEWORDER_QTY_INVALID,
"New Order Qty invalid must be numeric range 1 - 99."                           },
                {       ERR_NEWORDER_QTY_RANGE,
"New Order Qty is out of range. Range = 1 to 99."                               },
                {       ERR_NEWORDER_QTY_WITHOUT_SUPPW,                 "New
Order Qty field entered without a corresponding Supp_W."                        },
                {       ERR_NEWORDER_SUPPW_INVALID,
"New Order Supp_W invalid data type must be numeric."                           },
                {       ERR_NO_SERVER_SPECIFIED,                        "No
Server name specified."
                },
                {       ERR_ORDERSTATUS_CID_AND_CLT,                    "Order
Status Only Customer ID or Last Name may be entered, not both."         },
                {       ERR_ORDERSTATUS_CID_INVALID,                    "Order
Status Customer ID invalid, range must be numeric 1 - 3000."    },
                {       ERR_ORDERSTATUS_CLT_RANGE,
"Order Status Customer last name longer than 16 characters."            },
                {       ERR_ORDERSTATUS_DID_INVALID,                    "Order
Status District invalid, value must be numeric 1 - 10."                 },
                {       ERR_ORDERSTATUS_MISSING_CID_CLT,        "Order Status Either
Customer ID or Last Name must be entered."              },
                {       ERR_ORDERSTATUS_MISSING_CID_KEY,        "Order Status
missing Customer key \"CID*\"."                                         },
                {       ERR_ORDERSTATUS_MISSING_CLT_KEY,        "Order Status
missing Customer Last Name key \"CLT*\"."                               },
                {       ERR_ORDERSTATUS_MISSING_DID_KEY,        "Order Status
missing District key \"DID*\"."                                         },
                {       ERR_PAYMENT_CDI_INVALID,
"Payment Customer district invalid must be numeric."                            },
                {       ERR_PAYMENT_CID_AND_CLT,
"Payment Only Customer ID or Last Name may be entered, not both."       },
                {       ERR_PAYMENT_CUSTOMER_INVALID,
"Payment Customer data type invalid, must be numeric."                          },
                {       ERR_PAYMENT_CWI_INVALID,
"Payment Customer Warehouse invalid, must be numeric."                          },
                {       ERR_PAYMENT_DISTRICT_INVALID,
"Payment District ID is invalid, must be 1 - 10."                               },
                {       ERR_PAYMENT_HAM_INVALID,
"Payment Amount invalid data type must be numeric."                                     },
                {       ERR_PAYMENT_HAM_RANGE,
"Payment Amount out of range, 0 - 9999.99."
},
                {       ERR_PAYMENT_LAST_NAME_TO_LONG,
"Payment Customer last name longer than 16 characters."                 },
                {       ERR_PAYMENT_MISSING_CDI_KEY,
"Payment missing Customer district key \"CDI*\"."                       },
                {       ERR_PAYMENT_MISSING_CID_CLT,
"Payment Either Customer ID or Last Name must be entered."              },
                {       ERR_PAYMENT_MISSING_CID_KEY,
"Payment missing Customer Key \"CID*\"."
},
                {       ERR_PAYMENT_MISSING_CLT_KEY,
"Payment missing Customer Last Name key \"CLT*\"."                              },
                {       ERR_PAYMENT_MISSING_CWI_KEY,
"Payment missing Customer Warehouse key \"CWI*\"."                              },
                {       ERR_PAYMENT_MISSING_DID_KEY,
"Payment missing District Key \"DID*\"."
},
                {       ERR_PAYMENT_MISSING_HAM_KEY,
"Payment missing Amount key \"HAM*\"."
        },
                {       ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,   "Stock Level;
missing Threshold key \"TT*\"."                                         },
                {       ERR_STOCKLEVEL_THRESHOLD_INVALID,       "Stock Level;
Threshold value must be in the range = 1 - 99."                 },
                {       ERR_STOCKLEVEL_THRESHOLD_RANGE,                 "Stock
Level Threshold out of range, range must be 1 - 99."            },
                {       ERR_VERSION_MISMATCH,
"Invalid version field.  RTE and Web Client are probably out of sync." },
                {       ERR_W_ID_INVALID,
"Invalid Warehouse ID."
                },
                {       0,
                        ""

}
        };

        char szTmp[256];
        int i = 0;
        while (TRUE)
        {
                if (errorMsgs[i].szMsg[0] == 0)
                {
                        strcpy( szTmp, "Unknown error number." );
                        break;
                }
```

```
                        if (m_Error == errorMsgs[i].iError)
                        {
                                strcpy( szTmp, errorMsgs[i].szMsg );
                                break;
                        }
                        i++;
                }

                if (m_szTextDetail)
                        strcat( szTmp, m_szTextDetail );
                if (m_SystemErr)
                        wsprintf( szTmp+strlen(szTmp), "  Error=%d", m_SystemErr );

                m_szErrorText = new char[strlen(szTmp)+1];
                strcpy( m_szErrorText, szTmp );
                return m_szErrorText;

}

/* FUNCTION: GetKeyValue
 *
 * PURPOSE:             This function parses a http formatted string for specific key values.
 *
 * ARGUMENTS:           char                    *pQueryString       http string from client browser
 *                      char                    *pKey
key value to look for
 *                      char                    *pValue
character array into which to place key's value
 *                      int                             iMax
                maximum length of key value array.
 *                      WEBERROR                err
                error value to throw
 *
 * RETURNS:                     nothing.
 *
 * ERROR:               if (the pKey value is not found) then
 *                                              if (err == 0)
 *                                                      return (empty string)
 *                                              else
 *                                                      throw CWEBCLNT_ERR(err)
 *
 * COMMENTS:            http keys are formatted either KEY=value& or KEY=value\0. This DLL formats
 *                      TPC-C input fields in such a manner that the keys can be extracted in the
 *                      above manner.
 */

void GetKeyValue(char **pQueryString,char *pKey, char *pValue, int iMax, WEBERROR err)
{
        char *ptr;

        if ( !(ptr=strstr(*pQueryString,pKey)) )
                goto ErrorExit;
        ptr += strlen(pKey);
        if ( *ptr != '=' )
                goto ErrorExit;
        ptr++;
        iMax--;  // one position is for terminating null
        while( *ptr && *ptr != '&' && iMax)
        {
                *pValue++ = *ptr++;
                iMax--;
        }
        *pValue = 0;  // terminating null

        *pQueryString = ptr;
        return;

ErrorExit:
        if (err != NO_ERR)
                throw new CWEBCLNT_ERR( err );
        *pValue = 0;  // return empty result string
}

/* FUNCTION: GetIntKeyValue
 *
 * PURPOSE:             This function parses a http formatted string for a specific key value.
 *
 * ARGUMENTS:           char                    *pQueryString       http string from client browser
 *                      char                    *pKey
key value to look for
 *                      WEBERROR                NoKeyErr            error
value to throw if key not found
 *                      WEBERROR                NotIntErr           error
value to throw if value not numeric
 *
 * RETURNS:                     integer
 *
 * ERROR:               if (the pKey value is not found) then
 *                                              if (NoKeyErr != NO_ERR)
 *                                                      throw CWEBCLNT_ERR(err)
 *                                              else
 *                                                      return 0
 *                                      else if (non-numeric char found) then
 *                                              if (NotIntErr != NO_ERR) then
 *                                                      throw CWEBCLNT_ERR(err)
 *                                              else
 *                                                      return 0
 *
 * COMMENTS:            http keys are formatted either KEY=value& or KEY=value\0. This DLL formats
 *                                      TPC-C input fields in such a manner that the keys can be extracted in the
 *                                      above manner.
 */

int GetIntKeyValue(char **pQueryString,char *pKey, WEBERROR NoKeyErr, WEBERROR NotIntErr)
{
        char *ptr0;
        char *ptr;

        if ( !(ptr=strstr(*pQueryString,pKey)) )
                goto ErrorNoKey;
        ptr += strlen(pKey);
```

```
	if ( *ptr != '=' )
		goto ErrorNoKey;
	ptr++;

	ptr0 = ptr;			// remember starting point
	// scan string until a terminator (null or &) or a non-digit
	while( *ptr && *ptr != '&' && isdigit(*ptr) )
		ptr++;

	// make sure we stopped scanning for the right reason
	if ((ptr0 == ptr) || (*ptr && *ptr != '&'))
	{
		if (NotIntErr != NO_ERR)
			throw new CWEBCLNT_ERR( NoKeyErr );
		return 0;
	}

	*pQueryString = ptr;
	return atoi(ptr0);

ErrorNoKey:
	if (NoKeyErr != NO_ERR)
		throw new CWEBCLNT_ERR( NoKeyErr );
	return 0;
}

/* FUNCTION: TermInit
 *
 * PURPOSE:		This function initializes the client terminal structure; it is called when the TPCC.DLL
 *			is first loaded by the inet service.
 *
 */

void TermInit(void)
{
	EnterCriticalSection(&TermCriticalSection);

	Term.iMasterSyncId	= 1;
	Term.iNumEntries	= Reg.dwMaxConnections+1;

	Term.pClientData	= NULL;
	Term.pClientData	= (PCLIENTDATA)malloc(Term.iNumEntries* sizeof(CLIENTDATA));
	if (Term.pClientData == NULL)
	{
		LeaveCriticalSection(&TermCriticalSection);
		throw new CWEBCLNT_ERR( ERR_MEM_ALLOC_FAILED );
	}

	ZeroMemory( Term.pClientData, Term.iNumEntries * sizeof(CLIENTDATA) );

	Term.iFreeList			= Term.iNumEntries-1;
	// build free list
	// note: Term.pClientData[0].iNextFree gets set to -1, which marks it as "in use".
	//	This is intentional, as the zero entry is used as an anchor and never
	//			allocated as an actual terminal.
	for(int i=0; i<Term.iNumEntries; i++)
		Term.pClientData[i].iNextFree = i-1;

	LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermDeleteAll
 *
 * PURPOSE:		This function frees allocated resources associated with the terminal structure.
 *
 * ARGUMENTS:		none
 *
 * RETURNS:			None
 *
 * COMMENTS:		This function is called only when the inet service unloads the TPCC.DLL
 *
 */

void TermDeleteAll(void)
{
	EnterCriticalSection(&TermCriticalSection);

	for(int i=1; i<Term.iNumEntries; i++)
	{
		if (Term.pClientData[i].iNextFree == -1)
			delete Term.pClientData[i].pTxn;
	}

	Term.iFreeList			= 0;
	Term.iNumEntries	= 0;
	if ( Term.pClientData )
		free(Term.pClientData);
	Term.pClientData	= NULL;

	LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermAdd
 *
 * PURPOSE:		This function assigns a terminal id which is used to identify a client browser.
 *
 * RETURNS:			int			assigned terminal id
 *
 */

int TermAdd(void)
{
	DWORD	i;
	int			iNewTerm, iTickCount;

	if (Term.iNumEntries == 0)
		return -1;

	EnterCriticalSection(&TermCriticalSection);
	if (Term.iFreeList != 0)
	{	// position is available
		iNewTerm = Term.iFreeList;
```

```
                Term.iFreeList = Term.pClientData[iNewTerm].iNextFree;
                Term.pClientData[iNewTerm].iNextFree = -1;  // indicates this position is in use
        }
        else
        {       // no open slots, so find the slot that hasn't been used in the longest time and reuse it
                for(iNewTerm=1, i=1, iTickCount=0x7FFFFFFF;i<Reg.dwMaxConnections; i++)
                {
                        if (iTickCount > Term.pClientData[i].iTickCount)
                        {
                                iTickCount = Term.pClientData[i].iTickCount;
                                iNewTerm = i;
                        }
                }
                // if oldest term is less than one minute old, it probably means that more connections
                // are being attempted than were specified as "Max Connections" at install.  In this case,
                // do not bump existing connection; instead, return error torequestor.
                if ((GetTickCount() - iTickCount) < 60000)
                {
                        LeaveCriticalSection(&TermCriticalSection);
                        throw new CWEBCLNT_ERR( ERR_MAX_CONNECTIONS_EXCEEDED );
                }
        }

        Term.pClientData[iNewTerm].iTickCount= GetTickCount();
        Term.pClientData[iNewTerm].iSyncId= Term.iMasterSyncId++;
        Term.pClientData[iNewTerm].pTxn= NULL;

        LeaveCriticalSection(&TermCriticalSection);
        return iNewTerm;
}

/* FUNCTION: TermDelete
 *
 * PURPOSE:             This function makes a terminal entry in the Term array available for reuse.
 *
 * ARGUMENTS:           int                                                                     id
Terminal id of client exiting
 *
 */

void TermDelete(int id)
{
        if ( id > 0 && id < Term.iNumEntries)
        {
                delete Term.pClientData[id].pTxn;

                // put onto free list
                EnterCriticalSection(&TermCriticalSection);

                Term.pClientData[id].iNextFree= Term.iFreeList;
                Term.iFreeList= id;

                LeaveCriticalSection(&TermCriticalSection);
        }
}
```

```
/* FUNCTION: MakeErrorForm
 */

void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int iErrorNum, int iTermId, int iSyncId, char
*szErrorText, char *szBuffer )
{
        wsprintf(szBuffer,
                "<HTML><HEAD><TITLE>TPC-C Error</TITLE></HEAD><BODY>"
                "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
                "<BOLD>An Error Occurred</BOLD><BR><BR>"
                "%s"
                "<BR><BR><HR>"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
                "</FORM></BODY></HTML>"
                , iType, iErrorNum, MAIN_MENU_FORM, iTermId, iSyncId, szErrorText );
}

/* FUNCTION: MakeMainMenuForm
 */

void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm)
{
        wsprintf(szForm,
                "<HTML><HEAD><TITLE>TPC-C Main Menu</TITLE></HEAD><BODY>"
                "Select Desired Transaction.<BR><HR>"
                "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
                "</FORM></BODY></HTML>"
                , MAIN_MENU_FORM, iTermId, iSyncId);
}

/* FUNCTION: MakeStockLevelForm
 *
 * PURPOSE:             This function constructs the Stock LevelHTML page.
 *
```

```
 * COMMENTS:        The internal client buffer is created when the terminal id is assigned and should not
 *                                          be freed except when the client terminal id is no longer needed.
 */

void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA *pStockLevelData, BOOL bInput, char *szForm)
{
        int             c;

        c = wsprintf(szForm,
                        "<HTML><HEAD><TITLE>TPC-C Stock Level</TITLE></HEAD><FORM
ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<PRE><font face=\"Courier\">                           Stock-Level<BR>"
                        "Warehouse: %4.4d   District: %2.2d<BR> <BR>",
                        STOCK_LEVEL_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
                        Term.pClientData[iTermId].w_id, Term.pClientData[iTermId].d_id);

        if ( bInput )
        {
                strcpy(szForm+c,
                                "Stock Level Threshold: <INPUT NAME=\"TT*\" SIZE=2><BR> <BR>"
                                "low stock:    </font><BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR>"
                                " <BR> <BR> <BR> <BR> <BR> <BR> <BR></PRE><HR>"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                                "</FORM></HTML>" );
        }
        else
        {
                wsprintf(szForm+c,
                                "Stock Level Threshold: %2.2d<BR> <BR>"
                                "low stock: %3.3d</font> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR>"
                                " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR></PRE><HR>"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
                                "</FORM></HTML>"
                                , pStockLevelData->threshold, pStockLevelData->low_stock);
        }
}

/* FUNCTION: MakeNewOrderForm
 *
 * COMMENTS:        The internal client buffer is created when the terminal id is assigned and should not
 *                                          be freed except when the client terminal id is no longer needed.
 */

void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA *pNewOrderData, BOOL bInput, char *szForm)
{
        int                     i, c;
        BOOL    bValid;
        static  char szBR[] = " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR>";

        if (!bInput)
                assert( pNewOrderData->exec_status_code == eOK || pNewOrderData->exec_status_code ==
eInvalidItem );

        bValid = (bInput || (pNewOrderData->exec_status_code == eOK));

        c = wsprintf(szForm,
                "<HTML><HEAD><TITLE>TPC-C New Order</TITLE></HEAD><BODY>"
                "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
                "<PRE><font face=\"Courier\">                           New  Order<BR>"
                , bValid ? 0 : ERR_BAD_ITEM_ID, NEW_ORDER_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

        if ( bInput )
        {
                c += wsprintf(szForm+c, "Warehouse: %4.4d   ", Term.pClientData[iTermId].w_id);

                strcpy( szForm+c,
                        "District: <INPUT NAME=\"DID*\" SIZE=1>                 Date:<BR>"
                        "Customer:  <INPUT NAME=\"CID*\" SIZE=4>   Name:                Credit:
%Disc:<BR>"
                        "Order Number:          Number of Lines:        W_tax:      D_tax:<BR> <BR>"
                        " Supp_W  Item_Id  Item Name             Qty  Stock  B/G  Price    Amount<BR>"
                        "  <INPUT NAME=\"SP00*\" SIZE=4>  <INPUT NAME=\"IID00*\" SIZE=6>
<INPUT  NAME=\"Qty00*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP01*\" SIZE=4>  <INPUT NAME=\"IID01*\" SIZE=6>
<INPUT  NAME=\"Qty01*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP02*\" SIZE=4>  <INPUT NAME=\"IID02*\" SIZE=6>
<INPUT  NAME=\"Qty02*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP03*\" SIZE=4>  <INPUT NAME=\"IID03*\" SIZE=6>
<INPUT  NAME=\"Qty03*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP04*\" SIZE=4>  <INPUT NAME=\"IID04*\" SIZE=6>
<INPUT  NAME=\"Qty04*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP05*\" SIZE=4>  <INPUT NAME=\"IID05*\" SIZE=6>
<INPUT  NAME=\"Qty05*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP06*\" SIZE=4>  <INPUT NAME=\"IID06*\" SIZE=6>
<INPUT  NAME=\"Qty06*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP07*\" SIZE=4>  <INPUT NAME=\"IID07*\" SIZE=6>
<INPUT  NAME=\"Qty07*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP08*\" SIZE=4>  <INPUT NAME=\"IID08*\" SIZE=6>
<INPUT  NAME=\"Qty08*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP09*\" SIZE=4>  <INPUT NAME=\"IID09*\" SIZE=6>
<INPUT  NAME=\"Qty09*\" SIZE=1><BR>"
```

```
                        " <INPUT NAME=\"SP10*\" SIZE=4> <INPUT NAME=\"IID10*\" SIZE=6>
<INPUT NAME=\"Qty10*\" SIZE=1><BR>"
                        " <INPUT NAME=\"SP11*\" SIZE=4> <INPUT NAME=\"IID11*\" SIZE=6>
<INPUT NAME=\"Qty11*\" SIZE=1><BR>"
                        " <INPUT NAME=\"SP12*\" SIZE=4> <INPUT NAME=\"IID12*\" SIZE=6>
<INPUT NAME=\"Qty12*\" SIZE=1><BR>"
                        " <INPUT NAME=\"SP13*\" SIZE=4> <INPUT NAME=\"IID13*\" SIZE=6>
<INPUT NAME=\"Qty13*\" SIZE=1><BR>"
                        " <INPUT NAME=\"SP14*\" SIZE=4> <INPUT NAME=\"IID14*\" SIZE=6>
<INPUT NAME=\"Qty14*\" SIZE=1><BR>"
                        "Execution Status:                    Total:<BR>"
                        "</font></PRE><HR>"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                        "</FORM></HTML>"
                        );
        }
        else
        {
                c += wsprintf(szForm+c, "Warehouse: %4.4d   District: %2.2d                Date: ",
                        pNewOrderData->w_id,
                        pNewOrderData->d_id);

                if ( bValid )
                {
                        c += wsprintf(szForm+c, "%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
                                pNewOrderData->o_entry_d.day,
                                pNewOrderData->o_entry_d.month,
                                pNewOrderData->o_entry_d.year,
                                pNewOrderData->o_entry_d.hour,
                                pNewOrderData->o_entry_d.minute,
                                pNewOrderData->o_entry_d.second);
                }

                c += wsprintf(szForm+c, "<BR>Customer:  %4.4d   Name: %-16s   Credit: %-2s   ",
                        pNewOrderData->c_id, pNewOrderData->c_last, pNewOrderData->c_credit);

                if ( bValid )
                {
                        c += sprintf(szForm+c,
                                                        "%%Disc: %5.2f          <BR>"
                                                        "Order Number: %8.8d  Number of Lines:
%2.2d       W_tax: %5.2f   D_tax: %5.2f  <BR> <BR>"
                                                        " Supp_W  Item_Id  Item Name              Qty
Stock  B/G  Price    Amount<BR>",
                                100.0*pNewOrderData->c_discount,
                                pNewOrderData->o_id,
                                pNewOrderData->o_ol_cnt,
                                100.0 * pNewOrderData->w_tax,
                                100.0 * pNewOrderData->d_tax);

                        for(i=0; i<pNewOrderData->o_ol_cnt; i++)
                        {
                                c += sprintf(szForm+c," %4.4d   %6.6d   %-24s  %2.2d    %3.3d
%1.1s   $%6.2f  $%7.2f  <BR>",
                                                pNewOrderData->OL[i].ol_supply_w_id,
                                                pNewOrderData->OL[i].ol_i_id,
                                                pNewOrderData->OL[i].ol_i_name,
                                                pNewOrderData->OL[i].ol_quantity,
                                                pNewOrderData->OL[i].ol_stock,
                                                pNewOrderData->OL[i].ol_brand_generic,
                                                pNewOrderData->OL[i].ol_i_price,
                                                pNewOrderData->OL[i].ol_amount );
                        }
                }
                else
                {
                        c += wsprintf(szForm+c,
                                "%Disc:<BR>"
                                "Order Number: %8.8d  Number of Lines:          W_tax:
D_tax:<BR> <BR>"
                                " Supp_W  Item_Id  Item Name              Qty  Stock  B/G  Price
Amount<BR>"
                                , pNewOrderData->o_id);

                        i = 0;
                }

                strncpy( szForm+c, szBR, (15-i)*5 );
                c += (15-i)*5;

                if ( bValid )
                        c += sprintf(szForm+c, "Execution Status: Transaction commited.                Total:
$%8.2f  ",
                                pNewOrderData->total_amount);
                else
                        c += wsprintf(szForm+c, "Execution Status: Item number is not valid.
Total:");

                strcpy(szForm+c,
                        " <BR></font></PRE><HR>"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
                        "</FORM></HTML>"
                        );
        }
}

/* FUNCTION: MakePaymentForm
 *
 * COMMENTS:        The internal client buffer is created when the terminal id is assigned and should not
 *                                  be freed except when the client terminal id is no longer needed.
 */

void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData, BOOL bInput, char *szForm)
{
        int c;
```

```
        c = wsprintf(szForm,
                "<HTML><HEAD><TITLE>TPC-C Payment</TITLE></HEAD><BODY>"
                "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
                "<PRE><font face=\"Courier\">                                Payment<BR>"
                "Date: "
                , PAYMENT_FORM, iTermId, Term.pClientData[iTermId].iSyncId);

        if ( !bInput )
        {
                c += wsprintf(szForm+c, "%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
                        pPaymentData->h_date.day,
                        pPaymentData->h_date.month,
                        pPaymentData->h_date.year,
                        pPaymentData->h_date.hour,
                        pPaymentData->h_date.minute,
                        pPaymentData->h_date.second);
        }

        if ( bInput )
        {
                c += wsprintf(szForm+c,
                        "<BR> <BR>Warehouse: %4.4d"
                        "                District: <INPUT  NAME=\"DID*\" SIZE=1><BR> <BR> <BR>
<BR> <BR>"
                        "Customer: <INPUT NAME=\"CID*\" SIZE=4>"
                        "Cust-Warehouse: <INPUT NAME=\"CWI*\" SIZE=4>  "
                        "Cust-District: <INPUT NAME=\"CDI*\" SIZE=1><BR>"
                        "Name:                  <INPUT  NAME=\"CLT*\" SIZE=16>
Since:<BR>"
                        "                                Credit:<BR>"
                        "                                Disc:<BR>"
                        "                                Phone:<BR> <BR>"
                        "Amount Paid:         $<INPUT NAME=\"HAM*\" SIZE=7>      New
Cust-Balance:<BR>"
                        "Credit Limit:<BR> <BR>Cust-Data: <BR> <BR> <BR> <BR>
<BR></font></PRE><HR>"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\"><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                        "</BODY></FORM></HTML>"
                        , Term.pClientData[iTermId].w_id);
        }
        else
        {
                c += wsprintf(szForm+c,
                        "<BR> <BR>Warehouse: %4.4d                District:  %2.2d<BR>"
                        "%-20s                %-20s<BR>"
                        "%-20s                %-20s<BR>"
                        "%-20s %-2s %5.5s-%4.4s       %-20s %-2s %5.5s-%4.4s<BR> <BR>"
                        "Customer: %4.4d  Cust-Warehouse: %4.4d  Cust-District: %2.2d<BR>"
                        "Name:   %-16s %-2s %-16s     Since:  %2.2d-%2.2d-%4.4d<BR>"
                        "       %-20s                 Credit:  %-2s<BR>"
                        , Term.pClientData[iTermId].w_id, pPaymentData->d_id
                        , pPaymentData->w_street_1, pPaymentData->d_street_1
                        , pPaymentData->w_street_2, pPaymentData->d_street_2
                        , pPaymentData->w_city, pPaymentData->w_state, pPaymentData->w_zip,
pPaymentData->w_zip+5
                        , pPaymentData->d_city, pPaymentData->d_state, pPaymentData->d_zip,
pPaymentData->d_zip+5
                        , pPaymentData->c_id, pPaymentData->c_w_id,     pPaymentData->c_d_id
                        , pPaymentData->c_first, pPaymentData->c_middle, pPaymentData->c_last
                        , pPaymentData->c_since.day, pPaymentData->c_since.month,
pPaymentData->c_since.year
                        , pPaymentData->c_street_1, pPaymentData->c_credit
                        );

                c += sprintf(szForm+c,
                        "       %-20s                 %%Disc:  %5.2f<BR>",
                        pPaymentData->c_street_2, 100.0*pPaymentData->c_discount);

                c += wsprintf(szForm+c,
                        "       %-20s %-2s %5.5s-%4.4s       Phone:  %6.6s-%3.3s-%3.3s-%4.4s<BR>
<BR>",
                        pPaymentData->c_city, pPaymentData->c_state, pPaymentData->c_zip,
pPaymentData->c_zip+5,
                        pPaymentData->c_phone, pPaymentData->c_phone+6, pPaymentData->c_phone+9,
pPaymentData->c_phone+12 );

                c += sprintf(szForm+c,
                        "Amount Paid:         $%7.2f      New Cust-Balance: $%14.2f<BR>"
                        "Credit Limit:  $%13.2f<BR> <BR>"
                        , pPaymentData->h_amount, pPaymentData->c_balance
                        , pPaymentData->c_credit_lim
                        );

                if ( pPaymentData->c_credit[0] == 'B' && pPaymentData->c_credit[1] == 'C' )
                        c += wsprintf(szForm+c,
                                "Cust-Data: %-50.50s<BR>          %-50.50s<BR>
%-50.50s<BR>          %-50.50s<BR>",
                                pPaymentData->c_data, pPaymentData->c_data+50,
pPaymentData->c_data+100, pPaymentData->c_data+150 );
                else
                        strcpy(szForm+c, "Cust-Data: <BR> <BR> <BR> <BR>");

                strcat(szForm,                " <BR></font></PRE><HR>"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"
                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"
```

```
                                                                "</BODY></FORM></HTML>");
        }

}

/* FUNCTION: MakeOrderStatusForm
 *
 * COMMENTS:        The internal client buffer is created when the terminal id is assigned and should not
 *                                          be freed except when the client terminal id is no longer needed.
 */

void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA *pOrderStatusData, BOOL bInput, char *szForm)
{
        int                     i, c;
        static char szBR[] = " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR>";

        c = wsprintf(szForm,
                "<HTML><HEAD><TITLE>TPC-C Order-Status</TITLE></HEAD><BODY>"
                "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
                "<PRE><font face=\"Courier\">                   Order-Status<BR>"
                "Warehouse: %4.4d   ",
                ORDER_STATUS_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id);

        if ( bInput )
        {
                strcpy(szForm+c,
                        "District: <INPUT NAME=\"DID*\" SIZE=1><BR>"
                        "Customer: <INPUT NAME=\"CID*\" SIZE=4>   Name:                 <INPUT
NAME=\"CLT*\" SIZE=23><BR>"
                        "Cust-Balance:<BR><BR>"
                        "Order-Number:          Entry-Date:                 Carrier-Number:<BR>"
                        "Supply-W    Item-Id    Qty    Amount     Delivery-Date<BR><BR> <BR> <BR>
<BR>"
                        " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR></font></PRE>"
                        "<HR><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\"><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                        "</BODY></FORM></HTML>" );
        }
        else
        {
                c += wsprintf(szForm+c,
                        "District: %2.2d<BR>"
                        "Customer: %4.4d   Name: %-16s %-2s %-16s<BR>",
                        pOrderStatusData->d_id, pOrderStatusData->c_id,
                        pOrderStatusData->c_first, pOrderStatusData->c_middle, pOrderStatusData->c_last);

                c += sprintf(szForm+c, "Cust-Balance: $%9.2f<BR> <BR>",
                        pOrderStatusData->c_balance);

                c += wsprintf(szForm+c,
                        "Order-Number: %8.8d   Entry-Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d
Carrier-Number: %2.2d<BR>"
                        "Supply-W    Item-Id    Qty    Amount     Delivery-Date<BR>",
                        pOrderStatusData->o_id,
                        pOrderStatusData->o_entry_d.day,
                        pOrderStatusData->o_entry_d.month,
                        pOrderStatusData->o_entry_d.year,
                        pOrderStatusData->o_entry_d.hour,
                        pOrderStatusData->o_entry_d.minute,
                        pOrderStatusData->o_entry_d.second,
                        pOrderStatusData->o_carrier_id);

                for(i=0; i< pOrderStatusData->o_ol_cnt; i++)
                {
                        c += sprintf(szForm+c, "  %4.4d      %6.6d     %2.2d     $%8.2f
%2.2d-%2.2d-%4.4d<BR>",
                                pOrderStatusData->OL[i].ol_supply_w_id,
                                pOrderStatusData->OL[i].ol_i_id,
                                pOrderStatusData->OL[i].ol_quantity,
                                pOrderStatusData->OL[i].ol_amount,
                                pOrderStatusData->OL[i].ol_delivery_d.day,
                                pOrderStatusData->OL[i].ol_delivery_d.month,
                                pOrderStatusData->OL[i].ol_delivery_d.year);
                }

                strncpy( szForm+c, szBR, (15-i)*5 );
                c += (15-i)*5;

                strcpy(szForm+c,
                        "</font></PRE><HR><INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
                        "</BODY></FORM></HTML>" );
        }
}

/* FUNCTION: MakeDeliveryForm
 *
 * COMMENTS:        The internal client buffer is created when the terminal id is assigned and should not
 *                                          be freed except when the client terminal id is no longer needed.
 */

void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData, BOOL bInput, char *szForm)
{
        int             c;

        c = wsprintf(szForm,
                "<HTML><HEAD><TITLE>TPC-C Delivery</TITLE></HEAD><BODY>"
                "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
```

```
                        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<PRE><font face=\"Courier\">                          Delivery<BR>"
                        "Warehouse: %4.4d<BR> <BR>",
                        (!bInput && (pDeliveryData->exec_status_code != eOK)) ? ERR_TYPE_DELIVERY_POST : 0,
                        DELIVERY_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id);

        if ( bInput )
        {
                strcpy( szForm+c,
                        "Carrier Number: <INPUT NAME=\"OCD*\" SIZE=1><BR> <BR>"
                        "Execution Status: <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                        " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> </font></PRE><HR>"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                        "</BODY></FORM></HTML>" );
        }
        else
        {
                wsprintf( szForm+c,
                        "Carrier Number: %2.2d<BR> <BR>"
                        "Execution Status: %s <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                        " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> </font></PRE>"
                        "<HR><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
                        "</BODY></FORM></HTML>"

                        , pDeliveryData->o_carrier_id,
                        (pDeliveryData->exec_status_code == eOK) ? "Delivery has been queued." :
"Delivery Post Failed    "
                        );
        }
}

/* FUNCTION: ProcessNewOrderForm
 *
 * PURPOSE:             This function gets and validates the input data from the new order form
 *                              filling in the required input variables. it then calls theSQLNewOrder
 *                              transaction, constructs the output form and writes it back to client
 *                              browser.
 */

void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK            *pECB, int iTermId, char *szBuffer)
{
        PNEW_ORDER_DATA             pNewOrder;

        pNewOrder = Term.pClientData[iTermId].pTxn->BuffAddr_NewOrder();

        ZeroMemory(pNewOrder, sizeof(NEW_ORDER_DATA));
        pNewOrder->w_id = Term.pClientData[iTermId].w_id;
        GetNewOrderData(pECB->lpszQueryString, pNewOrder);

        Term.pClientData[iTermId].pTxn->NewOrder();

        pNewOrder = Term.pClientData[iTermId].pTxn->BuffAddr_NewOrder();
        MakeNewOrderForm(iTermId, pNewOrder, OUTPUT_FORM, szBuffer );
}

/* FUNCTION: void ProcessPaymentForm
 *
 * PURPOSE:             This function gets and validates the input data from the payment form
 *                              filling in the required input variables. It then calls theSQLPayment
 *                              transaction, constructs the output form and writes it back to client
 *                              browser.
 *
 * ARGUMENTS:           EXTENSION_CONTROL_BLOCK   *pECB       passed in structure pointer from inetsrv.
 *                                                      int
iTermId      client browser terminal id
 *
 */

void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK*pECB, int iTermId, char *szBuffer)
{
        PPAYMENT_DATA       pPayment;

        pPayment = Term.pClientData[iTermId].pTxn->BuffAddr_Payment();
        ZeroMemory(pPayment, sizeof(PAYMENT_DATA));
        pPayment->w_id = Term.pClientData[iTermId].w_id;
        GetPaymentData(pECB->lpszQueryString, pPayment);

        Term.pClientData[iTermId].pTxn->Payment();

        pPayment = Term.pClientData[iTermId].pTxn->BuffAddr_Payment();
        MakePaymentForm(iTermId, pPayment, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessOrderStatusForm
 *
 * PURPOSE:             This function gets and validates the input data from the Order Status
 *                              form filling in the required input variables. It then calls the
 *                              SQLOrderStatus transaction, constructs the output form and writes it
 *                              back to client browser.
 *
 * ARGUMENTS:           EXTENSION_CONTROL_BLOCK   *pECB       passed in structure pointer from inetsrv.
 *                                                      int
iTermId      client browser terminal id
 *
 */

void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer)
{
        PORDER_STATUS_DATA          pOrderStatus;

        pOrderStatus = Term.pClientData[iTermId].pTxn->BuffAddr_OrderStatus();
        ZeroMemory(pOrderStatus, sizeof(ORDER_STATUS_DATA));
```

```
		pOrderStatus->w_id = Term.pClientData[iTermId].w_id;
		GetOrderStatusData(pECB->lpszQueryString, pOrderStatus);

		Term.pClientData[iTermId].pTxn->OrderStatus();

		pOrderStatus = Term.pClientData[iTermId].pTxn->BuffAddr_OrderStatus();
		MakeOrderStatusForm(iTermId, pOrderStatus, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessDeliveryForm
 *
 * PURPOSE:		This function gets and validates the input data from the delivery form
 *				filling in the required input variables. It then calls thePostDeliveryInfo
 *				Api, The client is then informed that the transaction has been posted.
 *
 * ARGUMENTS:		EXTENSION_CONTROL_BLOCK	*pECB		passed in structure pointer from inetsrv.
 *				int
iTermId		client browser terminal id
 *
 */

void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK*pECB, int iTermId, char *szBuffer)
{
		char		*ptr = pECB->lpszQueryString;

		PDELIVERY_DATA	pDelivery;

		pDelivery = Term.pClientData[iTermId].pTxn->BuffAddr_Delivery();
		ZeroMemory(pDelivery, sizeof(DELIVERY_DATA));
		pDelivery->w_id = Term.pClientData[iTermId].w_id;

		pDelivery->o_carrier_id  = GetIntKeyValue(&ptr, "OCD*", ERR_DELIVERY_MISSING_OCD_KEY,
ERR_DELIVERY_CARRIER_INVALID);
		if ( pDelivery->o_carrier_id > 10 || pDelivery->o_carrier_id < 1 )
			throw new CWEBCLNT_ERR( ERR_DELIVERY_CARRIER_ID_RANGE );

		if (dwNumDeliveryThreads)
		{
			//post delivery info
			if ( PostDeliveryInfo(pDelivery->w_id,pDelivery->o_carrier_id) )
				pDelivery->exec_status_code = eDeliveryFailed;
			else
				pDelivery->exec_status_code = eOK;
		}
		else // delivery is done synchronously if no delivery threads configured
			Term.pClientData[iTermId].pTxn->Delivery();

		pDelivery = Term.pClientData[iTermId].pTxn->BuffAddr_Delivery();
		MakeDeliveryForm(iTermId,pDelivery, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessStockLevelForm
 *
 * PURPOSE:		This function gets and validates the input data from the Stock Level
 *				form filling in the required input variables. It then calls the
 *				SQLStockLevel transaction, constructs the output form and writes it
 *				back to client browser.
 *
 * ARGUMENTS:		EXTENSION_CONTROL_BLOCK	*pECB		passed in structure pointer from inetsrv.
 *				int
iTermId		client browser terminal id
 *
 */

void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK*pECB, int iTermId, char *szBuffer)
{
		char			*ptr = pECB->lpszQueryString;

		PSTOCK_LEVEL_DATA	pStockLevel;

		pStockLevel = Term.pClientData[iTermId].pTxn->BuffAddr_StockLevel();
		ZeroMemory( pStockLevel, sizeof(STOCK_LEVEL_DATA));

		pStockLevel->w_id = Term.pClientData[iTermId].w_id;
		pStockLevel->d_id = Term.pClientData[iTermId].d_id;

		pStockLevel->threshold = GetIntKeyValue(&ptr, "TT*",
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY, ERR_STOCKLEVEL_THRESHOLD_INVALID);
		if ( pStockLevel->threshold >= 100 || pStockLevel->threshold < 0 )
			throw new CWEBCLNT_ERR( ERR_STOCKLEVEL_THRESHOLD_RANGE );

		Term.pClientData[iTermId].pTxn->StockLevel();

		pStockLevel = Term.pClientData[iTermId].pTxn->BuffAddr_StockLevel();
		MakeStockLevelForm(iTermId, pStockLevel, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: GetNewOrderData
 *
 * PURPOSE:		This function extracts and validates the new order form data from anhttp command string.
 *
 * ARGUMENTS:		LPSTR			lpszQueryString		client browser http
command string
 *				NEW_ORDER_DATA	*pNewOrderData		pointer
to new order data structure
 *
 */

void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData)
{
		char		szTmp[26];
		int			i;
		short		items;
		int			ol_i_id, ol_quantity;
		char		*ptr = lpszQueryString;

		static char szSP[MAX_OL_NEW_ORDER_ITEMS][6] =
			{ "SP00*", "SP01*", "SP02*", "SP03*", "SP04*",
			  "SP05*", "SP06*", "SP07*", "SP08*", "SP09*",
			  "SP10*", "SP11*", "SP12*", "SP13*", "SP14*" };
		static char szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
			{ "IID00*", "IID01*", "IID02*", "IID03*", "IID04*",
```

```
                        "IID05*", "IID06*", "IID07*", "IID08*", "IID09*",
                        "IID10*", "IID11*", "IID12*", "IID13*", "IID14*" };
        static char szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
                        { "Qty00*", "Qty01*", "Qty02*", "Qty03*", "Qty04*",
                        "Qty05*", "Qty06*", "Qty07*", "Qty08*", "Qty09*",
                        "Qty10*", "Qty11*", "Qty12*", "Qty13*", "Qty14*" };

        pNewOrderData->d_id = GetIntKeyValue(&ptr, "DID*", ERR_NEWORDER_FORM_MISSING_DID,
ERR_NEWORDER_DISTRICT_INVALID);
        pNewOrderData->c_id = GetIntKeyValue(&ptr, "CID*", ERR_NEWORDER_CUSTOMER_KEY,
ERR_NEWORDER_CUSTOMER_INVALID);

        for(i=0, items=0; i<MAX_OL_NEW_ORDER_ITEMS; i++)
        {
                GetKeyValue(&ptr, szSP[i], szTmp, sizeof(szTmp),
ERR_NEWORDER_MISSING_SUPPW_KEY);
                if ( szTmp[0] )
                {
                        if ( !IsNumeric(szTmp))
                                throw new CWEBCLNT_ERR(
ERR_NEWORDER_SUPPW_INVALID );
                        pNewOrderData->OL[items].ol_supply_w_id = (short)atoi(szTmp);

                        ol_i_id = pNewOrderData->OL[items].ol_i_id =
                                GetIntKeyValue(&ptr, szIID[i],
ERR_NEWORDER_MISSING_IID_KEY, ERR_NEWORDER_ITEMID_INVALID);
                        if ( ol_i_id > 999999 || ol_i_id < 1 )
                                throw new CWEBCLNT_ERR( ERR_NEWORDER_ITEMID_RANGE
);

                        ol_quantity = pNewOrderData->OL[items].ol_quantity=
                                GetIntKeyValue(&ptr, szQty[i],
ERR_NEWORDER_MISSING_QTY_KEY, ERR_NEWORDER_QTY_INVALID);
                        if ( ol_quantity > 99 || ol_quantity < 1 )
                                throw new CWEBCLNT_ERR( ERR_NEWORDER_QTY_RANGE );

                        items++;
                }
                else
                {       // nothing entered for supply warehouse, so item id and qty must also be blank
                        GetKeyValue(&ptr, szIID[i], szTmp, sizeof(szTmp),
ERR_NEWORDER_MISSING_IID_KEY);
                        if ( szTmp[0] )
                                throw new CWEBCLNT_ERR(
ERR_NEWORDER_ITEMID_WITHOUT_SUPPW );

                        GetKeyValue(&ptr, szQty[i], szTmp, sizeof(szTmp),
ERR_NEWORDER_MISSING_QTY_KEY);
                        if ( szTmp[0] )
                                throw new CWEBCLNT_ERR(
ERR_NEWORDER_QTY_WITHOUT_SUPPW );
                }
        }
        if ( items == 0 )
                throw new CWEBCLNT_ERR( ERR_NEWORDER_NOITEMS_ENTERED );

        pNewOrderData->o_ol_cnt = items;
}

/* FUNCTION: GetPaymentData
 *
 * PURPOSE:             This function extracts and validates the payment form data from anhttp command string.
 *
 * ARGUMENTS:   LPSTR                   lpszQueryString                 client browser http
command string
 *                                      PAYMENT_DATA    *pPaymentData                   pointer
to payment data structure
 */

void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData)
{
        char    szTmp[26];
        char    *ptr = lpszQueryString;
        BOOL    bCustIdBlank;

        pPaymentData->d_id = GetIntKeyValue(&ptr, "DID*", ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_DISTRICT_INVALID);

        GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp), ERR_PAYMENT_MISSING_CID_KEY);
        if ( szTmp[0] == 0 )
        {
                bCustIdBlank = TRUE;
                pPaymentData->c_id = 0;
        }
        else
        {       // parse customer id and verify that last name was NOT entered
                bCustIdBlank = FALSE;
                if ( !IsNumeric(szTmp))
                        throw new CWEBCLNT_ERR( ERR_PAYMENT_CUSTOMER_INVALID );
                pPaymentData->c_id = atoi(szTmp);
        }

        pPaymentData->c_w_id = GetIntKeyValue(&ptr, "CWI*", ERR_PAYMENT_MISSING_CWI_KEY,
ERR_PAYMENT_CWI_INVALID);
        pPaymentData->c_d_id = GetIntKeyValue(&ptr, "CDI*", ERR_PAYMENT_MISSING_CDI_KEY,
ERR_PAYMENT_CDI_INVALID);

        if ( bCustIdBlank )
        {       // customer id is blank, so last name must be entered
                GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
                if ( szTmp[0] == 0 )
                        throw new CWEBCLNT_ERR( ERR_PAYMENT_MISSING_CID_CLT );

                _strupr( szTmp );
                if ( strlen(pPaymentData->c_last) > LAST_NAME_LEN )
                        throw new CWEBCLNT_ERR( ERR_PAYMENT_LAST_NAME_TO_LONG );
                strcpy(pPaymentData->c_last, szTmp);
        }
        else
        {       // parse customer id and verify that last name was NOT entered
                GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
                if ( szTmp[0] != 0 )
```

```
                        throw new CWEBCLNT_ERR( ERR_PAYMENT_CID_AND_CLT );
        }

        GetKeyValue(&ptr, "HAM*", szTmp, sizeof(szTmp), ERR_PAYMENT_MISSING_HAM_KEY);
        if (!IsDecimal(szTmp))
                throw new CWEBCLNT_ERR( ERR_PAYMENT_HAM_INVALID );
        pPaymentData->h_amount = atof(szTmp);
        if ( pPaymentData->h_amount >= 10000.00 || pPaymentData->h_amount < 0 )
                throw new CWEBCLNT_ERR( ERR_PAYMENT_HAM_RANGE );
}

/* FUNCTION: GetOrderStatusData
 *
 * PURPOSE:             This function extracts and validates the payment form data from anhttp command string.
 *
 */
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA *pOrderStatusData)
{
        char            szTmp[26];
        char            *ptr = lpszQueryString;

        pOrderStatusData->d_id = GetIntKeyValue(&ptr, "DID*", ERR_ORDERSTATUS_MISSING_DID_KEY,
ERR_ORDERSTATUS_DID_INVALID);

        GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CID_KEY);
        if ( szTmp[0] == 0 )
        {       // customer id is blank, so last name must be entered
                pOrderStatusData->c_id = 0;
                GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
                if ( szTmp[0] == 0 )
                        throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_MISSING_CID_CLT );

                _strupr( szTmp );
                if ( strlen(pOrderStatusData->c_last) > LAST_NAME_LEN )
                        throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CLT_RANGE );
                strcpy(pOrderStatusData->c_last, szTmp);
        }
        else
        {       // parse customer id and verify that last name was NOT entered
                if ( !IsNumeric(szTmp) )
                        throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CID_INVALID );
                pOrderStatusData->c_id = atoi(szTmp);
                GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
                if ( szTmp[0] != 0 )
                        throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CID_AND_CLT );
        }

}

/* FUNCTION: BOOL IsNumeric(char *ptr)
 *
 * PURPOSE:             This function determines if a string is numeric. It fails if any characters other
 *                              than numeric and null terminator are present.
 *
 * ARGUMENTS:           char                            *ptr            pointer to string to check.
 *
 * RETURNS:                     BOOL    FALSE   if string is not all numeric
 *                                                      TRUE    if string contains only numeric
characters i.e. '0' - '9'
 */

BOOL IsNumeric(char *ptr)
{
        if ( *ptr == 0 )
                return FALSE;

        while( *ptr && isdigit(*ptr) )
                ptr++;
        return ( !*ptr );
}

/* FUNCTION: BOOL IsDecimal(char *ptr)
 *
 * PURPOSE:             This function determines if a string is a non-negative decimal value.
 *      It fails if any characters other than a series of numbers followed by
 *                              a decimal point, another series of numbers, and a null terminator are present.
 *
 * ARGUMENTS:           char                            *ptr            pointer to string to check.
 *
 * RETURNS:                     BOOL    FALSE   if string is not a valid non-negative decimal value
 *                                                      TRUE    if string is OK
 */

BOOL IsDecimal(char *ptr)
{
        char *dotptr;
        BOOL  bValid;

        if ( *ptr == 0 )
                return FALSE;

        // find decimal point
        dotptr = strchr( ptr, '.' );
        if (dotptr == NULL)
                // no decimal point, so just check for numeric
                return IsNumeric(ptr);
        *dotptr = 0;  // temporarily replace decimal with a terminator

        if ( *ptr != 0 )
                bValid = IsNumeric(ptr);
        // string starts with decimal point
        else if (*(dotptr+1) == 0)
                return FALSE;  // nothing but a decimal point is bad
        else
                bValid = TRUE;

        if (*(dotptr+1) != 0)
                // check text after decimal point
                bValid &= IsNumeric(dotptr+1);
```

```
        *dotptr = '.';  // replace decimal point
        return bValid;
}
```

tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

        GetExtensionVersion     @1
        HttpExtensionProc       @2
        TerminateExtension      @3
```

tpcc.h

```
/*      FILE:           TPCC.H
 *                              Microsoft TPC-C Kit Ver. 4.20.000
 *                              Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                              Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 *      PURPOSE:  Header file for ISAPI TPCC.DLL, defines structures and functions used in the isapi tpcc.dll.
 *
 */

//VERSION RESOURCE DEFINES
#define _APS_NEXT_RESOURCE_VALUE                101
#define _APS_NEXT_COMMAND_VALUE                 40001
#define _APS_NEXT_CONTROL_VALUE                 1000
#define _APS_NEXT_SYMED_VALUE                   101

#define TP_MAX_RETRIES                                  50


//note that the welcome form must be processed first as terminal ids assigned here, once the
//terminal id is assigned then the forms can be processed in any order.
#define WELCOME_FORM                            1
//beginning form no term id assigned, form id
#define MAIN_MENU_FORM                          2
//term id assigned main menu form id
#define NEW_ORDER_FORM                          3
//new order form id
#define PAYMENT_FORM                            4
//payment form id
#define DELIVERY_FORM                           5
//delivery form id
#define ORDER_STATUS_FORM               6               //order
status id
#define STOCK_LEVEL_FORM                7               //stock
level form id

//This macro is used to prevent the compiler error unused formal parameter
#define UNUSEDPARAM(x) (x = x)


//This structure defines the data necessary to keep distinct for each terminal or client connection.
typedef struct _CLIENTDATA
{
        int                     iNextFree;              //index of next free
element or -1 if this entry in use.
        int                     w_id;
//warehouse id assigned at welcome form
        int                     d_id;                   //district
id assigned at welcome form

        int                     iSyncId;                //syncronization id
        int                     iTickCount;             //time of last access;

        CTPCC_BASE      *pTxn;

} CLIENTDATA, *PCLIENTDATA;


//This structure is used to define the operational interface for terminal id support
typedef struct _TERM
{
        int                     iNumEntries;
//total allocated terminal array entries
        int                     iFreeList;
//next available terminal array element or -1 if none
        int                     iMasterSyncId;
//syncronization id
        CLIENTDATA      *pClientData;           //pointer
to allocated client data
} TERM;

typedef TERM *PTERM;                            //pointer to terminal
structure type


        enum WEBERROR
        {
                NO_ERR,
                ERR_COMMAND_UNDEFINED,
                ERR_D_ID_INVALID,
                ERR_DELIVERY_CARRIER_ID_RANGE,
                ERR_DELIVERY_CARRIER_INVALID,
                ERR_DELIVERY_MISSING_OCD_KEY,
                ERR_DELIVERY_THREAD_FAILED,
                ERR_GETPROCADDR_FAILED,
                ERR_HTML_ILL_FORMED,
                ERR_INVALID_SYNC_CONNECTION,
                ERR_INVALID_TERMID,
                ERR_LOADDLL_FAILED,
```

```
                ERR_MAX_CONNECTIONS_EXCEEDED,
                ERR_MEM_ALLOC_FAILED,
                ERR_MISSING_REGISTRY_ENTRIES,
                ERR_NEWORDER_CUSTOMER_INVALID,
                ERR_NEWORDER_CUSTOMER_KEY,
                ERR_NEWORDER_DISTRICT_INVALID,
                ERR_NEWORDER_FORM_MISSING_DID,
                ERR_NEWORDER_ITEMID_INVALID,
                ERR_NEWORDER_ITEMID_RANGE,
                ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
                ERR_NEWORDER_MISSING_IID_KEY,
                ERR_NEWORDER_MISSING_QTY_KEY,
                ERR_NEWORDER_MISSING_SUPPW_KEY,
                ERR_NEWORDER_NOITEMS_ENTERED,
                ERR_NEWORDER_QTY_INVALID,
                ERR_NEWORDER_QTY_RANGE,
                ERR_NEWORDER_QTY_WITHOUT_SUPPW,
                ERR_NEWORDER_SUPPW_INVALID,
                ERR_NO_SERVER_SPECIFIED,
                ERR_ORDERSTATUS_CID_AND_CLT,
                ERR_ORDERSTATUS_CID_INVALID,
                ERR_ORDERSTATUS_CLT_RANGE,
                ERR_ORDERSTATUS_DID_INVALID,
                ERR_ORDERSTATUS_MISSING_CID_CLT,
                ERR_ORDERSTATUS_MISSING_CID_KEY,
                ERR_ORDERSTATUS_MISSING_CLT_KEY,
                ERR_ORDERSTATUS_MISSING_DID_KEY,
                ERR_PAYMENT_CDI_INVALID,
                ERR_PAYMENT_CID_AND_CLT,
                ERR_PAYMENT_CUSTOMER_INVALID,
                ERR_PAYMENT_CWI_INVALID,
                ERR_PAYMENT_DISTRICT_INVALID,
                ERR_PAYMENT_HAM_INVALID,
                ERR_PAYMENT_HAM_RANGE,
                ERR_PAYMENT_LAST_NAME_TO_LONG,
                ERR_PAYMENT_MISSING_CDI_KEY,
                ERR_PAYMENT_MISSING_CID_CLT,
                ERR_PAYMENT_MISSING_CID_KEY,
                ERR_PAYMENT_MISSING_CLT,
                ERR_PAYMENT_MISSING_CLT_KEY,
                ERR_PAYMENT_MISSING_CWI_KEY,
                ERR_PAYMENT_MISSING_DID_KEY,
                ERR_PAYMENT_MISSING_HAM_KEY,
                ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
                ERR_STOCKLEVEL_THRESHOLD_INVALID,
                ERR_STOCKLEVEL_THRESHOLD_RANGE,
                ERR_VERSION_MISMATCH,
                ERR_W_ID_INVALID
        };

class CWEBCLNT_ERR : public CBaseErr
{
        public:

                CWEBCLNT_ERR(WEBERROR Err)
                {
                        m_Error = Err;
                        m_szTextDetail = NULL;
                        m_SystemErr = 0;
                        m_szErrorText = NULL;
                };

                CWEBCLNT_ERR(WEBERROR Err, char *szTextDetail, DWORD dwSystemErr)
                {
                        m_Error = Err;
                        m_szTextDetail = new char[strlen(szTextDetail)+1];
                        strcpy( m_szTextDetail, szTextDetail );
                        m_SystemErr = dwSystemErr;
                        m_szErrorText = NULL;
                };

                ~CWEBCLNT_ERR()
                {
                        if (m_szTextDetail != NULL)
                                delete [] m_szTextDetail;
                        if (m_szErrorText != NULL)
                                delete [] m_szErrorText;
                };

                WEBERROR        m_Error;
                char            *m_szTextDetail;        //
                char            *m_szErrorText;
                DWORD           m_SystemErr;

                int ErrorType() {return ERR_TYPE_WEBDLL;};
                int ErrorNum() {return m_Error;};
                char *ErrorText();

};

//These constants have already been defined in engstut.h, but since we do
//not want to include it in the delisrv executable
#define     TXN_EVENT_START             2
#define     TXN_EVENT_STOP              4
#define TXN_EVENT_WARNING   6                   //used to record a warning into the log

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved);
void WriteMessageToEventLog(LPTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int *pFormId, int *pTermId, int
*pSyncId);
void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int iErrorType, char *szMsg, int iTermId);
void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int iMax, WEBERROR err);
int  GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR NoKeyErr, WEBERROR NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
```

```
int  TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int iErrorNum, int iTermId, int iSyncId, char
*szErrorText, char *szBuffer );
void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA *pStockLevelData, BOOL bInput, char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA *pNewOrderData, BOOL bInput, char *szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData, BOOL bInput, char *szForm);
void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA *pOrderStatusData, BOOL bInput, char *szForm);
void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData, BOOL bInput, char *szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK                *pECB, int iTermId, char *szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK*pECB, int iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK*pECB, int iTermId, char *szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK*pECB, int iTermId, char *szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK*pECB, int iTermId, char *szBuffer);
void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData);
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA *pOrderStatusData);
BOOL PostDeliveryInfo(short w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);
```

tpcc.rc

```
//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "afxres.h"

////////////////////////////////////////////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

////////////////////////////////////////////////////////////////////////////
// English (U.S.) resources

#if !defined(AFX_RESOURCE_DLL)|| defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif //_WIN32

#ifndef _MAC
////////////////////////////////////////////////////////////////////////////
//
// Version
//
VS_VERSION_INFO VERSIONINFO
 FILEVERSION 0,4,0,0
 PRODUCTVERSION 0,4,0,0
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x40004L
 FILETYPE 0x2L
 FILESUBTYPE 0x0L
BEGIN
   BLOCK "StringFileInfo"
   BEGIN
      BLOCK "040904b0"
      BEGIN
         VALUE "Comments", "TPC-C HTML DLL Server (DBLIB)\0"
         VALUE "CompanyName", "Microsoft\0"
         VALUE "FileDescription","TPC-C HTML DLL Server (DBLIB)\0"
         VALUE "FileVersion", "0, 4, 0, 0\0"
         VALUE "InternalName", "tpcc\0"
         VALUE "LegalCopyright","Copyright © 1997\0"
         VALUE "OriginalFilename","tpcc.dll\0"
         VALUE "ProductName", "Microsoft tpcc\0"
         VALUE "ProductVersion", "0, 4, 0, 0\0"
      END
   END
   BLOCK "VarFileInfo"
   BEGIN
      VALUE "Translation", 0x409, 1200
   END
END

#endif    // !_MAC


#ifdef APSTUDIO_INVOKED
////////////////////////////////////////////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
   "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
   "#include ""afxres.h""\r\n"
   "\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
```

```
    "\r\n"
    "\0"
END

#endif    // APSTUDIO_INVOKED


////////////////////////////////////////////////////////////////////////////
//
// Dialog
//

IDD_DIALOG1 DIALOG DISCARDABLE  0, 0, 186, 95
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Dialog"
FONT 8, "MS Sans Serif"
BEGIN
    DEFPUSHBUTTON   "OK",IDOK,129,7,50,14
    PUSHBUTTON      "Cancel",IDCANCEL,129,24,50,14
END


////////////////////////////////////////////////////////////////////////////
//
// DESIGNINFO
//

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 179
        TOPMARGIN, 7
        BOTTOMMARGIN, 88
    END
END
#endif    // APSTUDIO_INVOKED

#endif    // English (U.S.) resources
////////////////////////////////////////////////////////////////////////////



#ifndef APSTUDIO_INVOKED
////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//


////////////////////////////////////////////////////////////////////////////
#endif    // not APSTUDIO_INVOKED
```

tpcc_com.cpp

```
/*              FILE:                   TPCC_COM.CPP
 *                                              Microsoft TPC-C Kit Ver. 4.20.000
 *                                              Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                              not yet audited
 *
 *              PURPOSE:  Source file for TPC-C COM+ class implementation.
 *              Contact:        Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                          4.20.000 - first version
 */

// needed for CoinitializeEx
#define _WIN32_WINNT 0x0400

#include <windows.h>

// need to declare functions for export
#define DllDecl  __declspec( dllexport )

#include "..\..\common\src\trans.h"                     //tpckit transaction header contains definations of structures specific to
TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_com.h"

#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\tpcc_com_all\src\tpcc_com_all_i.c"

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_COM* CTPCC_COM_new(BOOL bSinglePool)
{
        return new CTPCC_COM(bSinglePool);
}

CTPCC_COM::CTPCC_COM(BOOL bSinglePool)
{
        HRESULT hr = NULL;
        long lRet = 0;

        m_bSinglePool = bSinglePool;

        m_pNewOrder                     = NULL;
        m_pPayment              = NULL;
        m_pStockLevel           = NULL;
        m_pOrderStatus          = NULL;

        m_pTxn = (COM_DATA*)CoTaskMemAlloc(sizeof(COM_DATA));
        if (!m_pTxn)
                throw new CCOMERR( E_FAIL );
```

```
	hr = CoInitializeEx(NULL,COINIT_MULTITHREADED);
	if (FAILED(hr))
	{
		throw new CCOMERR( hr );
	}

	// create components
	if (m_bSinglePool)
	{
		hr = CoCreateInstance(CLSID_TPCC, NULL, CLSCTX_SERVER, IID_ITPCC, (void
**)&m_pNewOrder);
		if (FAILED(hr))
			throw new CCOMERR(hr);

		// all txns will use same component
		m_pPayment = m_pNewOrder;
		m_pStockLevel = m_pNewOrder;
		m_pOrderStatus = m_pNewOrder;
	}
	else
	{
		// use different components for each txn

		hr = CoCreateInstance(CLSID_NewOrder, NULL, CLSCTX_SERVER, IID_ITPCC, (void
**)&m_pNewOrder);
		if (FAILED(hr))
			throw new CCOMERR(hr);

		hr = CoCreateInstance(CLSID_Payment, NULL, CLSCTX_SERVER, IID_ITPCC, (void
**)&m_pPayment);
		if (FAILED(hr))
			throw new CCOMERR(hr);

		hr = CoCreateInstance(CLSID_StockLevel, NULL, CLSCTX_SERVER, IID_ITPCC, (void
**)&m_pStockLevel);
		if (FAILED(hr))
			throw new CCOMERR(hr);

		hr = CoCreateInstance(CLSID_OrderStatus, NULL, CLSCTX_SERVER, IID_ITPCC, (void
**)&m_pOrderStatus);
		if (FAILED(hr))
			throw new CCOMERR(hr);
	}

	// call setcomplete to release each component back into pool
	hr = m_pNewOrder->CallSetComplete();
	if (FAILED(hr))
		throw new CCOMERR(hr);

	if (!m_bSinglePool)
	{
		hr = m_pPayment->CallSetComplete();
		if (FAILED(hr))
			throw new CCOMERR(hr);

		hr = m_pStockLevel->CallSetComplete();
		if (FAILED(hr))
			throw new CCOMERR(hr);

		hr = m_pOrderStatus->CallSetComplete();
		if (FAILED(hr))
			throw new CCOMERR(hr);
	}
}

CTPCC_COM::~CTPCC_COM()
{
	if (m_pTxn)
		CoTaskMemFree(m_pTxn);

	ReleaseInterface(m_pNewOrder);
	if (!m_bSinglePool)
	{
		ReleaseInterface(m_pPayment);
		ReleaseInterface(m_pStockLevel);
		ReleaseInterface(m_pOrderStatus);
	}
	CoUninitialize();
}

void CTPCC_COM::NewOrder()
{
	int iSize = sizeof(COM_DATA);

	HRESULT hr = m_pNewOrder->NewOrder(&iSize, (unsigned char**)&m_pTxn);
	if (FAILED(hr))
		throw new CCOMERR( hr );

	if ( m_pTxn->ErrorType != ERR_SUCCESS )
		throw new CCOMERR( m_pTxn->ErrorType, m_pTxn->error );
}

void CTPCC_COM::Payment()
{
	int iSize = sizeof(COM_DATA);

	HRESULT hr = m_pPayment->Payment(&iSize,(unsigned char**)&m_pTxn);
	if (FAILED(hr))
		throw new CCOMERR( hr );

	if ( m_pTxn->ErrorType != ERR_SUCCESS )
		throw new CCOMERR( m_pTxn->ErrorType, m_pTxn->error );
}

void CTPCC_COM::StockLevel()
{
	int iSize = sizeof(COM_DATA);

	HRESULT hr = m_pStockLevel->StockLevel(&iSize,(unsigned char**)&m_pTxn);
	if (FAILED(hr))
		throw new CCOMERR( hr );
```

```
	if ( m_pTxn->ErrorType != ERR_SUCCESS )
		throw new CCOMERR( m_pTxn->ErrorType, m_pTxn->error );
}

void CTPCC_COM::OrderStatus()
{
	int iSize = sizeof(COM_DATA);

	HRESULT hr = m_pOrderStatus->OrderStatus(&iSize, (unsigned char**)&m_pTxn);
	if (FAILED(hr))
		throw new CCOMERR( hr );

	if ( m_pTxn->ErrorType != ERR_SUCCESS )
		throw new CCOMERR( m_pTxn->ErrorType, m_pTxn->error );
}
```

tpcc_com.h

```
/*	FILE:		TPCC_COM.H
 *					Microsoft TPC-C Kit Ver. 4.20.000
 *					Copyright Microsoft, 1999
 *	All Rights Reserved
 *
 *					not yet audited
 *
 *	PURPOSE:  Header file for TPC-C COM+ class implementation.
 *
 *  Change history:
 *		4.20.000 - first version
 */

#pragma once

#include <stdio.h>
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class CCOMERR : public CBaseErr
{
	private:
		char m_szErrorText[64];

	public:
		// use this interface for genuine COM errors
		CCOMERR( HRESULT hr )
		{
			m_hr = hr;
			m_iErrorType = 0;
			m_iError = 0;
		}

		// use this interface to impersonate a non-COM error type
		CCOMERR( int iErrorType, int iError )
		{
			m_iErrorType = iErrorType;
			m_iError = iError;
			m_hr = S_OK;
		}

		int		m_hr;
		int		m_iErrorType;
		int		m_iError;

		// A CCOMERR class can impersonate another class, which happens if the error
		// was not actually a COM Services error, but was simply transmitted back via COM.
		int ErrorType()
		{
			if (m_iErrorType == 0)
				return ERR_TYPE_COM;
			else
				return m_iErrorType;
		}

		int ErrorNum() {return m_hr;}

		char *ErrorText()
		{
			if (m_hr == S_OK)
				sprintf( m_szErrorText, "Error: Class %d, error # %d", m_iErrorType,
m_iError );
			else
				sprintf( m_szErrorText, "Error: COM HRESULT %x", m_hr );
			return m_szErrorText;
		}
};

class DllDecl CTPCC_COM : public CTPCC_BASE
{
	private:
		BOOL m_bSinglePool;

		// COM Interface pointers
		ITPCC*		m_pNewOrder;
		ITPCC*		m_pPayment;
		ITPCC*		m_pStockLevel;
		ITPCC*		m_pOrderStatus;

		struct COM_DATA
		{
			int ErrorType;
			int error;
			union
			{
				NEW_ORDER_DATA		NewOrder;
				PAYMENT_DATA		Payment;
```

```
                                        DELIVERY_DATA           Delivery;
                                        STOCK_LEVEL_DATA StockLevel;
                                        ORDER_STATUS_DATA       OrderStatus;
                                } u;
                        } *m_pTxn;

        public:
                        CTPCC_COM(BOOL bSinglePool);
                        ~CTPCC_COM(void);

                        inline PNEW_ORDER_DATA          BuffAddr_NewOrder()             { return
&m_pTxn->u.NewOrder;    };
                        inline PPAYMENT_DATA            BuffAddr_Payment()              { return
&m_pTxn->u.Payment;     };
                        inline PDELIVERY_DATA           BuffAddr_Delivery()             { return
&m_pTxn->u.Delivery;    };
                        inline PSTOCK_LEVEL_DATA        BuffAddr_StockLevel()   { return
&m_pTxn->u.StockLevel; };
                        inline PORDER_STATUS_DATA       BuffAddr_OrderStatus()  { return
&m_pTxn->u.OrderStatus;};

                        void NewOrder                   ();
                        void Payment                    ();
                        void StockLevel                 ();
                        void OrderStatus        ();
                        void Delivery                   () { throw new CCOMERR(E_NOTIMPL); }  // not
supported
};


inline void ReleaseInterface(IUnknown *pUnk)
{
        if (pUnk)
        {
                pUnk->Release();
                pUnk = NULL;
        }
}


// wrapper routine for class constructor
extern "C" __declspec(dllexport) CTPCC_COM* CTPCC_COM_new(BOOL);

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);
```

tpcc_com_all.cpp

```
/*      FILE:           TPCC_COM_ALL.CPP
 *                      Microsoft TPC-C Kit Ver. 4.20.000
 *                      Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                      Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 *      PURPOSE:  Implementation for TPC-C Tuxedo class.
 *      Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

#include <stdio.h>
#include <atlbase.h>
//You may derive a class from CComModule and use it if you want to override
//something, but do not change the name of _Module
extern CComModule _Module;

#include <atlcom.h>
#include <initguid.h>
#include <transact.h>
#include <atlimpl.cpp>
#include <comsvcs.h>

#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"                                     //tpckit transaction header
contains definations of structures specific to TPC-C
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\error.h"
#include "..\..\common\src\ReadRegistry.h"
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"          // DBLIB implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"            // ODBC implementation of TPC-C txns


#include "resource.h"
#include "tpcc_com_all.h"
#include "tpcc_com_all_i.c"
#include "Methods.h"
#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\common\src\ReadRegistry.cpp"


CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
        OBJECT_ENTRY(CLSID_TPCC, CTPCC)
        OBJECT_ENTRY(CLSID_NewOrder, CNewOrder)
        OBJECT_ENTRY(CLSID_OrderStatus, COrderStatus)
        OBJECT_ENTRY(CLSID_Payment, CPayment)
        OBJECT_ENTRY(CLSID_StockLevel, CStockLevel)
END_OBJECT_MAP()
```

```
// configuration settings from registry
TPCCREGISTRYDATA Reg;
char                                    szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

static  HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB   *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC                 *pCTPCC_ODBC_new;


////////////////////////////////////////////////////////////////////////
// DLL Entry Point

extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID /*lpReserved*/)
{
        char szDllName[128];

        try
        {
                if (dwReason == DLL_PROCESS_ATTACH)
                {
                        _Module.Init(ObjectMap, hInstance);
                        DisableThreadLibraryCalls(hInstance);

                        DWORD dwSize = MAX_COMPUTERNAME_LENGTH+1;
                        GetComputerName(szMyComputerName, &dwSize);
                        szMyComputerName[dwSize] = 0;

                        if ( ReadTPCCRegistrySettings( &Reg ) )
                                throw new CCOMPONENT_ERR(
ERR_MISSING_REGISTRY_ENTRIES );

                        if (Reg.eDB_Protocol == DBLIB)
                        {
                                strcpy( szDllName, Reg.szPath );
                                strcat( szDllName, "tpcc_dblib.dll");
                                hLibInstanceDb = LoadLibrary( szDllName );
                                if (hLibInstanceDb == NULL)
                                        throw new CCOMPONENT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                                // get function pointer to wrapper for class constructor
                                pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
                                if (pCTPCC_DBLIB_new == NULL)
                                        throw new CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                        }
                        else if (Reg.eDB_Protocol == ODBC)
                        {
                                strcpy( szDllName, Reg.szPath );
                                strcat( szDllName, "tpcc_odbc.dll");
                                hLibInstanceDb = LoadLibrary( szDllName );
                                if (hLibInstanceDb == NULL)
                                        throw new CCOMPONENT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );

                                // get function pointer to wrapper for class constructor
                                pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                if (pCTPCC_ODBC_new == NULL)
                                        throw new CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                        }
                        else
                                throw new CCOMPONENT_ERR(
ERR_UNKNOWN_DB_PROTOCOL );
                }
                else if (dwReason == DLL_PROCESS_DETACH)
                        _Module.Term();

        }
        catch (CBaseErr *e)
        {
                WriteMessageToEventLog(e->ErrorText());
                delete e;
                return FALSE;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception in object DllMain"));
                return FALSE;
        }

        return TRUE;            // OK
}

////////////////////////////////////////////////////////////////////////
// Used to determine whether the DLL can be unloaded by OLE

STDAPI DllCanUnloadNow(void)
{
        return (_Module.GetLockCount()==0) ? S_OK : S_FALSE;
}

////////////////////////////////////////////////////////////////////////
// Returns a class factory to create an object of the requested type

STDAPI DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID* ppv)
{
        return _Module.GetClassObject(rclsid, riid, ppv);
}

////////////////////////////////////////////////////////////////////////
// DllRegisterServer - Adds entries to the system registry

STDAPI DllRegisterServer(void)
{
        // registers object, typelib and all interfaces in typelib
        return _Module.RegisterServer(TRUE);
```

```
}

////////////////////////////////////////////////////////////////////////////
// DllUnregisterServer - Removes entries from the system registry

STDAPI DllUnregisterServer(void)
{
        _Module.UnregisterServer();
        return S_OK;
}


static void WriteMessageToEventLog(LPTSTR lpszMsg)
{
   TCHAR   szMsg[256];
   HANDLE  hEventSource;
   LPTSTR  lpszStrings[2];

   // Use event logging to log the error.
   //
   hEventSource = RegisterEventSource(NULL, TEXT("tpcc_com_all.dll"));

   _stprintf(szMsg, TEXT("Error in COM+ TPC-C Component: "));
   lpszStrings[0] = szMsg;
   lpszStrings[1] = lpszMsg;

   if (hEventSource != NULL)
        {
      ReportEvent(hEventSource, // handle of event source
         EVENTLOG_ERROR_TYPE,  // event type
         0,                    // event category
         0,                    // event ID
         NULL,                 // current user's SID
         2,                    // strings in  lpszStrings
         0,                    // no bytes of raw data
         (LPCTSTR *)lpszStrings,    // array of error strings
         NULL);                // no raw data

      (VOID) DeregisterEventSource(hEventSource);
   }
}

inline void ReleaseInterface(IUnknown *pUnk)
{
        if (pUnk)
        {
                pUnk->Release();
                pUnk = NULL;
        }
}

/* FUNCTION: CCOMPONENT_ERR::ErrorText
 *
 */

char* CCOMPONENT_ERR::ErrorText(void)
{
        static SERRORMSG errorMsgs[] =
        {
                { ERR_MISSING_REGISTRY_ENTRIES,         "Required entries missing from registry."
                        },
                { ERR_LOADDLL_FAILED,                           "Load of DLL failed.  DLL="
                                                                },
                { ERR_GETPROCADDR_FAILED,               "Could not map proc in DLL.  GetProcAddr
error.  DLL="           },
                { ERR_UNKNOWN_DB_PROTOCOL,                      "Unknown database protocol
specified in registry."                 },
                { 0,                                                                                    ""

                        }
        };

        char szTmp[256];
        int i = 0;
        while (TRUE)
        {
                if (errorMsgs[i].szMsg[0] == 0)
                {
                        strcpy( szTmp, "Unknown error number." );
                        break;
                }
                if (m_Error == errorMsgs[i].iError)
                {
                        strcpy( szTmp, errorMsgs[i].szMsg );
                        break;
                }
                i++;
        }

        if (m_szTextDetail)
                strcat( szTmp, m_szTextDetail );
        if (m_SystemErr)
                wsprintf( szTmp+strlen(szTmp), "  Error=%d", m_SystemErr );

        m_szErrorText = new char[strlen(szTmp)+1];
        strcpy( m_szErrorText, szTmp );
        return m_szErrorText;
}


CTPCC_Common::CTPCC_Common()
{
        m_pTxn = NULL;
        m_bCanBePooled = TRUE;
}

CTPCC_Common::~CTPCC_Common()
{
        if (m_pTxn)
                delete m_pTxn;
}
```

```
HRESULT CTPCC_Common::CallSetComplete()
{
	IObjectContext* pObjectContext = NULL;

	// get our object context
	HRESULT hr = CoGetObjectContext( IID_IObjectContext, (void **)&pObjectContext );
	pObjectContext->SetComplete();
	ReleaseInterface(pObjectContext);
	return hr;
}

//
//	called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch* pUnk)
{
	// Code to access construction string, if needed later...
	//	if (!pUnk)
	//		return E_UNEXPECTED;
	//	IObjectConstructString * pString = NULL;
	//	HRESULT hr = pUnk->QueryInterface(IID_IObjectConstructString,(void **)&pString);
	//	pString->Release();

	try
	{
		if (Reg.eDB_Protocol == ODBC)
			m_pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
		else if (Reg.eDB_Protocol == DBLIB)
			m_pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
	}
	catch (CBaseErr *e)
	{
		WriteMessageToEventLog(e->ErrorText());
		delete e;
		return E_FAIL;
	}
	catch (...)
	{
		WriteMessageToEventLog(TEXT("Unhandledexception in object ::Construct"));
		return E_FAIL;
	}

	return S_OK;
}

HRESULT CTPCC_Common::NewOrder(int*iSize, UCHAR **txn)
{
	PNEW_ORDER_DATA pNewOrder;
	COM_DATA			*pData;

	try
	{
		pData = (COM_DATA*)*txn;
		pNewOrder = m_pTxn->BuffAddr_NewOrder();
		memcpy(pNewOrder, &pData->u.NewOrder, sizeof(NEW_ORDER_DATA));
		m_pTxn->NewOrder();
		memcpy( &pData->u.NewOrder, pNewOrder, sizeof(NEW_ORDER_DATA));

		pData->retval = ERR_SUCCESS;
		pData->error = 0;
		return S_OK;
	}
	catch (CBaseErr *e)
	{
		// check for lost database connection; if yes, component is toast
		if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
			((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
			m_bCanBePooled = FALSE;

		pData->retval = e->ErrorType();
		pData->error = e->ErrorNum();
		delete e;
		return E_FAIL;
	}
	catch (...)
	{
		WriteMessageToEventLog(TEXT("Unhandledexception."));
		pData->retval = ERR_TYPE_LOGIC;
		pData->error = 0;
		m_bCanBePooled = FALSE;
		return E_FAIL;
	}
}

HRESULT CTPCC_Common::Payment(int*iSize, UCHAR** txn)
{
	PPAYMENT_DATA	pPayment;
	COM_DATA			*pData;

	try
	{
		pData = (COM_DATA*)*txn;
		pPayment = m_pTxn->BuffAddr_Payment();

		memcpy(pPayment, &pData->u.Payment, sizeof(PAYMENT_DATA) );
		m_pTxn->Payment();
		memcpy( &pData->u.Payment, pPayment, sizeof(PAYMENT_DATA) );

		pData->retval = ERR_SUCCESS;
		pData->error = 0;
		return S_OK;
	}
	catch (CBaseErr *e)
	{
		// check for lost database connection; if yes, component is toast
		if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
			((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
			m_bCanBePooled = FALSE;
```

```
                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}

HRESULT CTPCC_Common::StockLevel(int*iSize, UCHAR** txn)
{
        PSTOCK_LEVEL_DATA       pStockLevel;
        COM_DATA                *pData;

        try
        {
                pData = (COM_DATA*)*txn;
                pStockLevel = m_pTxn->BuffAddr_StockLevel();

                memcpy(pStockLevel, &pData->u.StockLevel, sizeof(STOCK_LEVEL_DATA));
                m_pTxn->StockLevel();
                memcpy( &pData->u.StockLevel, pStockLevel, sizeof(STOCK_LEVEL_DATA));

                pData->retval = ERR_SUCCESS;
                pData->error = 0;
                return S_OK;
        }
        catch (CBaseErr *e)
        {
                // check for lost database connection; if yes, component is toast
                if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                        ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                        m_bCanBePooled = FALSE;

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}

HRESULT CTPCC_Common::OrderStatus(int*iSize, UCHAR** txn)
{
        PORDER_STATUS_DATA      pOrderStatus;
        COM_DATA                *pData;

        try
        {
                pData = (COM_DATA*)*txn;
                pOrderStatus = m_pTxn->BuffAddr_OrderStatus();

                memcpy(pOrderStatus, &pData->u.OrderStatus, sizeof(ORDER_STATUS_DATA));
                m_pTxn->OrderStatus();
                memcpy( &pData->u.OrderStatus, pOrderStatus, sizeof(ORDER_STATUS_DATA));

                pData->retval = ERR_SUCCESS;
                pData->error = 0;
                return S_OK;
        }
        catch (CBaseErr *e)
        {
                // check for lost database connection; if yes, component is toast
                if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                        ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                        m_bCanBePooled = FALSE;

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}
```

tpcc_com_all.def

```
; tpcc_com_all.def : Declares the module parameters.

LIBRARY      "tpcc_com_all.dll"

EXPORTS
        DllCanUnloadNow     @1 PRIVATE
        DllGetClassObject   @2 PRIVATE
        DllRegisterServer   @3 PRIVATE
        DllUnregisterServer         @4 PRIVATE
```

tpcc_com_all.dsp

```
# Microsoft Developer Studio Project File - Name="tpcc_com_all" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=tpcc_com_all - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_all.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_all.mak" CFG="tpcc_com_all - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_all - Win32 Release" (based on\
 "Win32 (x86) Dynamic-Link Library")
!MESSAGE "tpcc_com_all - Win32 Debug" (based on\
 "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tpcc_com_all - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib..\db_odbc_dll\bin\tpcc_odbc.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /machine:I386

!ELSEIF  "$(CFG)" == "tpcc_com_all - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib..\db_odbc_dll\bin\tpcc_odbc.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /pdbtype:sept

!ENDIF

# Begin Target

# Name "tpcc_com_all - Win32 Release"
# Name "tpcc_com_all - Win32 Debug"
# Begin Group "Source"

# PROP Default_Filter "*.cpp, *.c"
# Begin Source File

SOURCE=.\src\tpcc_com_all.cpp

!IF  "$(CFG)" == "tpcc_com_all - Win32 Release"

# SUBTRACT CPP /YX

!ELSEIF  "$(CFG)" == "tpcc_com_all - Win32 Debug"
```

```
# SUBTRACT CPP /FA<none> /YX

!ENDIF

# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_all.def
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_all.idl

!IF  "$(CFG)" == "tpcc_com_all - Win32 Release"

# Begin Custom Build - Performing MIDL step
InputPath=.\src\tpcc_com_all.idl

BuildCmds= \
            midl /Oicf /h "tpcc_com_all.h" /iid "tpcc_com_all_i.c"\
    ".\src\tpcc_com_all.idl"   /out ".\src"

".\src\tpcc_com_all.tlb" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)

".\src\tpcc_com_all.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)

".\src\tpcc_com_all_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)
# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_all - Win32 Debug"

# Begin Custom Build - Performing MIDL step
InputPath=.\src\tpcc_com_all.idl

BuildCmds= \
            midl /Oicf /h "tpcc_com_all.h" /iid "tpcc_com_all_i.c"\
    ".\src\tpcc_com_all.idl"   /out ".\src"

".\src\tpcc_com_all.tlb" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)

".\src\tpcc_com_all.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)

".\src\tpcc_com_all_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=.\src\Methods.h
# End Source File
# Begin Source File

SOURCE=.\src\resource.h
# End Source File
# End Group
# Begin Source File

SOURCE=.\src\tpcc_com_all.rc

!IF  "$(CFG)" == "tpcc_com_all - Win32 Release"

!ELSEIF  "$(CFG)" == "tpcc_com_all - Win32 Debug"

!ENDIF

# End Source File
# End Target
# End Project
```

tpcc_com_all.h

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the interfaces*/


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:24 1999
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )


/* verify that the <rpcndr.h> version is high enough to compile thisfile*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifndef __tpcc_com_all_h__
```

```
#define __tpcc_com_all_h__

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

#ifdef __cplusplus
typedef class TPCC TPCC;
#else
typedef struct TPCC TPCC;
#endif /* __cplusplus */

#endif          /* __TPCC_FWD_DEFINED__ */


#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

#ifdef __cplusplus
typedef class NewOrder NewOrder;
#else
typedef struct NewOrder NewOrder;
#endif /* __cplusplus */

#endif          /* __NewOrder_FWD_DEFINED__ */


#ifndef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

#ifdef __cplusplus
typedef class OrderStatus OrderStatus;
#else
typedef struct OrderStatus OrderStatus;
#endif /* __cplusplus */

#endif          /* __OrderStatus_FWD_DEFINED__ */


#ifndef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

#ifdef __cplusplus
typedef class Payment Payment;
#else
typedef struct Payment Payment;
#endif /* __cplusplus */

#endif          /* __Payment_FWD_DEFINED__ */


#ifndef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

#ifdef __cplusplus
typedef class StockLevel StockLevel;
#else
typedef struct StockLevel StockLevel;
#endif /* __cplusplus */

#endif          /* __StockLevel_FWD_DEFINED__ */


/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"
#include "tpcc_com_ps.h"

#ifdef __cplusplus
extern "C"{
#endif

void __RPC_FAR * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void __RPC_FAR * );

/* interface __MIDL_itf_tpcc_com_all_0000 */
/* [local] */




extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;


#ifndef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

/* library TPCCLib */
/* [helpstring][version][uuid]*/


EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus

class DECLSPEC_UUID("122A3128-2520-11D3-BA71-00C04FBFE08B")
TPCC;
#endif

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

class DECLSPEC_UUID("975BAABF-84A7-11D2-BA47-00C04FBFE08B")
```

```
NewOrder;
#endif

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

class DECLSPEC_UUID("266836AD-A50D-11D2-BA4E-00C04FBFE08B")
OrderStatus;
#endif

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus

class DECLSPEC_UUID("CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B")
Payment;
#endif

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-00C04FBFE08B")
StockLevel;
#endif
#endif /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces*/

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_all.idl

```
/*          FILE:                   TPCC.IDL
 *                                  Microsoft TPC-C Kit Ver. 4.20.000
 *                                  Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                                  not yet audited
 *
 *          PURPOSE:  IDL source for TPCC.dll.  This file is processed by the MIDL tool to
 *                                  produce the type library (TPCC.tlb) and marshalling code.
 *
 *  Change history:
 *                      4.20.000 - first version
 */
```

```
interface TPCC;
interface NewOrder;
interface OrderStatus;
interface Payment;
interface StockLevel;


import "oaidl.idl";
import "ocidl.idl";
import "..\tpcc_com_ps\src\tpcc_com_ps.idl";

[
        uuid(122A3117-2520-11D3-BA71-00C04FBFE08B),
        version(1.0),
        helpstring("TPC-C 1.0 Type Library")
]
library TPCCLib
{
        importlib("stdole32.tlb");
        importlib("stdole2.tlb");

        [
                uuid(122A3128-2520-11D3-BA71-00C04FBFE08B),
                helpstring("All Txns Class")
        ]
        coclass TPCC
        {
                [default] interface ITPCC;
        };


        [
                uuid(975BAABF-84A7-11D2-BA47-00C04FBFE08B),
                helpstring("NewOrder Class")
        ]
        coclass NewOrder
        {
                [default] interface ITPCC;
        };


        [
                uuid(266836AD-A50D-11D2-BA4E-00C04FBFE08B),
                helpstring("OrderStatus Class")
        ]
        coclass OrderStatus
        {
                [default] interface ITPCC;
        };


        [
                uuid(CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B),
                helpstring("Payment Class")
        ]
        coclass Payment
```

```
	{
		[default] interface ITPCC;
	};


	[
		uuid(2668369E-A50D-11D2-BA4E-00C04FBFE08B),
		helpstring("StockLevel Class")
	]
	coclass StockLevel
	{
		[default] interface ITPCC;
	};

};
```

tpcc_com_all.rc

```
//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "winres.h"

////////////////////////////////////////////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

////////////////////////////////////////////////////////////////////////////
// English (U.S.) resources

#if !defined(AFX_RESOURCE_DLL)|| defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif //_WIN32

#ifdef APSTUDIO_INVOKED
////////////////////////////////////////////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
   "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
   "#include ""winres.h""\r\n"
   "\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
   "1 TYPELIB ""tpcc_com_all.tlb""\r\n"
   "\0"
END

#endif    // APSTUDIO_INVOKED


#ifndef _MAC
////////////////////////////////////////////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
 FILEVERSION 1,0,0,1
 PRODUCTVERSION 1,0,0,1
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x4L
 FILETYPE 0x2L
 FILESUBTYPE 0x0L
BEGIN
   BLOCK "StringFileInfo"
   BEGIN
      BLOCK "040904B0"
      BEGIN
         VALUE "CompanyName", "\0"
         VALUE "FileDescription", "tpcc_com_all Module\0"
         VALUE "FileVersion", "1, 0, 0, 1\0"
         VALUE "InternalName", "TPCCNEWORDER\0"
         VALUE "LegalCopyright", "Copyright 1997\0"
         VALUE "OriginalFilename", "tpcc_com_all.DLL\0"
         VALUE "ProductName", "tpcc_com_all Module\0"
         VALUE "ProductVersion", "1, 0, 0, 1\0"
         VALUE "OLESelfRegister", "\0"
      END
   END
   BLOCK "VarFileInfo"
   BEGIN
      VALUE "Translation", 0x409, 1200
   END
END

#endif    // !_MAC
```

```
//////////////////////////////////////////////////////////////////////////////
//
// REGISTRY
//

IDR_TPCC              REGISTRY DISCARDABLE    "tpcc_com_all.rgs"
IDR_NEWORDER          REGISTRY DISCARDABLE    "tpcc_com_no.rgs"
IDR_ORDERSTATUS       REGISTRY DISCARDABLE    "tpcc_com_os.rgs"
IDR_PAYMENT           REGISTRY DISCARDABLE    "tpcc_com_pay.rgs"
IDR_STOCKLEVEL        REGISTRY DISCARDABLE    "tpcc_com_sl.rgs"

//////////////////////////////////////////////////////////////////////////////
//
// String Table
//

STRINGTABLE DISCARDABLE
BEGIN
   IDS_PROJNAME           "tpcc_com_all"
END

#endif    // English (U.S.) resources
//////////////////////////////////////////////////////////////////////////////



#ifndef APSTUDIO_INVOKED
//////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
1 TYPELIB "tpcc_com_all.tlb"

//////////////////////////////////////////////////////////////////////////////
#endif    // not APSTUDIO_INVOKED
```

tpcc_com_all.rgs

```
HKCR
{
	TPCC.AllTxns.1 = s 'All Txns Class'
	{
		CLSID = s '{122A3128-2520-11D3-BA71-00C04FBFE08B}'
	}
	TPCC.AllTxns = s 'TPCC Class'
	{
		CurVer = s 'TPCC.AllTxns.1'
	}
	NoRemove CLSID
	{
		ForceRemove {122A3128-2520-11D3-BA71-00C04FBFE08B} = s 'TPCC Class'
		{
			ProgID = s 'TPCC.AllTxns.1'
			VersionIndependentProgID = s 'TPCC.AllTxns'
			InprocServer32 = s '%MODULE%'
			{
				val ThreadingModel = s 'Both'
			}
		}
	}
}
```

tpcc_com_all_i.c

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients*/


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:24 1999
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_
```

```
#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
   unsigned long x;
   unsigned short s1;
   unsigned short s2;
   unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* !defined(_M_IA64) && !defined(_M_AXP64)*/
```

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients*/


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:25 1999
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if defined(_M_IA64) || defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
   unsigned long x;
   unsigned short s1;
   unsigned short s2;
   unsigned char  c[8];
} IID;
```

```
#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* defined(_M_IA64) || defined(_M_AXP64)*/
```

methods.h

```
/*              FILE:                   METHODS.H
 *                                      Microsoft TPC-C Kit Ver. 4.20.000
 *                                      Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                      not yet audited
 *
 *              PURPOSE:  Header file for COM components.
 *
 *  Change history:
 *                          4.20.000 - first version
 */


enum COMPONENT_ERROR
{
        ERR_MISSING_REGISTRY_ENTRIES = 1,
        ERR_LOADDLL_FAILED,
        ERR_GETPROCADDR_FAILED,
        ERR_UNKNOWN_DB_PROTOCOL
};

class CCOMPONENT_ERR : public CBaseErr
{
        public:
                CCOMPONENT_ERR(COMPONENT_ERROR Err)
                {
                        m_Error = Err;
                        m_szTextDetail = NULL;
                        m_SystemErr = 0;
                        m_szErrorText = NULL;
                };

                CCOMPONENT_ERR(COMPONENT_ERROR Err, char *szTextDetail, DWORD dwSystemErr)
                {
                        m_Error = Err;
                        m_szTextDetail = new char[strlen(szTextDetail)+1];
                        strcpy( m_szTextDetail, szTextDetail );
                        m_SystemErr = dwSystemErr;
                        m_szErrorText = NULL;
                };

                ~CCOMPONENT_ERR()
                {
                        if (m_szTextDetail != NULL)
                                delete [] m_szTextDetail;
                        if (m_szErrorText != NULL)
                                delete [] m_szErrorText;
                };

                COMPONENT_ERROR m_Error;
                char                            *m_szTextDetail;
                char                            *m_szErrorText;
                DWORD                           m_SystemErr;

                int ErrorType() {return ERR_TYPE_COMPONENT;};
                int ErrorNum() {return m_Error;};
                char *ErrorText();
};


static void WriteMessageToEventLog(LPTSTR lpszMsg);
```

```
/////////////////////////////////////////////////////////////////////////////
// CTPCC_Common
class CTPCC_Common :
	public ITPCC,
	public IObjectControl,
	public IObjectConstruct,
	public CComObjectRootEx<CComSingleThreadModel>
{
public:
BEGIN_COM_MAP(CTPCC_Common)
	COM_INTERFACE_ENTRY(ITPCC)
	COM_INTERFACE_ENTRY(IObjectControl)
	COM_INTERFACE_ENTRY(IObjectConstruct)
END_COM_MAP()

	CTPCC_Common();
	~CTPCC_Common();

// ITPCC
public:
	HRESULT __stdcall NewOrder(		int* iSize, UCHAR** txn);
	HRESULT __stdcall Payment(		int* iSize, UCHAR** txn);
	HRESULT __stdcall Delivery(		int* iSize, UCHAR** txn) {return E_NOTIMPL;}
	HRESULT __stdcall StockLevel(	int* iSize, UCHAR** txn);
	HRESULT __stdcall OrderStatus(	int* iSize, UCHAR** txn);

	HRESULT __stdcall CallSetComplete();

// IObjectControl
	STDMETHODIMP_(BOOL) CanBePooled() { return m_bCanBePooled; }
	STDMETHODIMP Activate() { return S_OK; }	// we don't support COM Services transactions (no
enlistment)
	STDMETHODIMP_(void) Deactivate() { /* nothing to do */ }

// IObjectConstruct
	STDMETHODIMP Construct(IDispatch * pUnk);

	// helper methods
private:
	BOOL		m_bCanBePooled;
	CTPCC_BASE	*m_pTxn;

	struct COM_DATA
	{
		int retval;
		int error;
		union
		{
			NEW_ORDER_DATA		NewOrder;
			PAYMENT_DATA		Payment;
			DELIVERY_DATA		Delivery;
			STOCK_LEVEL_DATA	StockLevel;
			ORDER_STATUS_DATA	OrderStatus;
		} u;
	};

};

/////////////////////////////////////////////////////////////////////////////
// CTPCC
class CTPCC :
	public CTPCC_Common,
	public CComCoClass<CTPCC, &CLSID_TPCC>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_TPCC)

BEGIN_COM_MAP(CTPCC)
	COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
	COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

};


/////////////////////////////////////////////////////////////////////////////
// CNewOrder
class CNewOrder :
	public CTPCC_Common,
	public CComCoClass<CNewOrder, &CLSID_NewOrder>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_NEWORDER)

BEGIN_COM_MAP(CNewOrder)
	COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
	COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
//	HRESULT __stdcall NewOrder(		int* iSize, UCHAR** txn) {return E_NOTIMPL;}
	HRESULT __stdcall Payment(		int* iSize, UCHAR** txn) {return E_NOTIMPL;}
	HRESULT __stdcall StockLevel(	int* iSize, UCHAR** txn) {return E_NOTIMPL;}
	HRESULT __stdcall OrderStatus(	int* iSize, UCHAR** txn) {return E_NOTIMPL;}
};


	/////////////////////////////////////////////////////////////////////////////
// COrderStatus
class COrderStatus :
	public CTPCC_Common,
	public CComCoClass<COrderStatus, &CLSID_OrderStatus>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_ORDERSTATUS)

BEGIN_COM_MAP(COrderStatus)
	COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
	COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
```

```
END_COM_MAP()

// ITPCC
public:
        HRESULT __stdcall NewOrder(             int* iSize, UCHAR** txn) {return E_NOTIMPL;}
        HRESULT __stdcall Payment(              int* iSize, UCHAR** txn) {return E_NOTIMPL;}
        HRESULT __stdcall StockLevel(   int* iSize, UCHAR** txn) {return E_NOTIMPL;}
//      HRESULT __stdcall OrderStatus(  int* iSize, UCHAR** txn) {return E_NOTIMPL;}
};


/////////////////////////////////////////////////////////////////////////////
// CPayment
class CPayment :
        public CTPCC_Common,
        public CComCoClass<CPayment, &CLSID_Payment>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_PAYMENT)

BEGIN_COM_MAP(CPayment)
        COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
        HRESULT __stdcall NewOrder(             int* iSize, UCHAR** txn) {return E_NOTIMPL;}
//      HRESULT __stdcall Payment(              int* iSize, UCHAR** txn) {return E_NOTIMPL;}
        HRESULT __stdcall StockLevel(   int* iSize, UCHAR** txn) {return E_NOTIMPL;}
        HRESULT __stdcall OrderStatus(  int* iSize, UCHAR** txn) {return E_NOTIMPL;}
};


/////////////////////////////////////////////////////////////////////////////
// CStockLevel
class CStockLevel :
        public CTPCC_Common,
        public CComCoClass<CStockLevel, &CLSID_StockLevel>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_STOCKLEVEL)

BEGIN_COM_MAP(CStockLevel)
        COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
        HRESULT __stdcall NewOrder(             int* iSize, UCHAR** txn) {return E_NOTIMPL;}
        HRESULT __stdcall Payment(              int* iSize, UCHAR** txn) {return E_NOTIMPL;}
//      HRESULT __stdcall StockLevel(   int* iSize, UCHAR** txn) {return E_NOTIMPL;}
        HRESULT __stdcall OrderStatus(  int* iSize, UCHAR** txn) {return E_NOTIMPL;}
};
```

tpcc_com_no.rgs

```
HKCR
{
        TPCC.NewOrder.1 = s 'NewOrder Class'
        {
                CLSID = s '{975BAABF-84A7-11D2-BA47-00C04FBFE08B}'
        }
        TPCC.NewOrder = s 'NewOrder Class'
        {
                CurVer = s 'TPCC.NewOrder.1'
        }
        NoRemove CLSID
        {
                ForceRemove {975BAABF-84A7-11D2-BA47-00C04FBFE08B} = s 'NewOrder Class'
                {
                        ProgID = s 'TPCC.NewOrder.1'
                        VersionIndependentProgID = s 'TPCC.NewOrder'
                        InprocServer32 = s '%MODULE%'
                        {
                                val ThreadingModel = s 'Both'
                        }
                }
        }
}
```

tpcc_com_os.rgs

```
HKCR
{
        TPCC.OrderStatus.1 = s 'OrderStatus Class'
        {
                CLSID = s '{266836AD-A50D-11D2-BA4E-00C04FBFE08B}'
        }
        TPCC.OrderStatus = s 'OrderStatus Class'
        {
                CurVer = s 'TPCC.OrderStatus.1'
        }
        NoRemove CLSID
        {
                ForceRemove {266836AD-A50D-11D2-BA4E-00C04FBFE08B} = s 'OrderStatus Class'
                {
                        ProgID = s 'TPCC.OrderStatus.1'
                        VersionIndependentProgID = s 'TPCC.OrderStatus'
                        InprocServer32 = s '%MODULE%'
                        {
                                val ThreadingModel = s 'Both'
                        }
                }
        }
}
```

tpcc_com_pay.rgs

```
HKCR
{
	TPCC.Payment.1 = s 'Payment Class'
	{
		CLSID = s '{CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B}'
	}
	TPCC.Payment = s 'Payment Class'
	{
		CurVer = s 'TPCC.Payment.1'
	}
	NoRemove CLSID
	{
		ForceRemove {CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B} = s 'Payment Class'
		{
			ProgID = s 'TPCC.Payment.1'
			VersionIndependentProgID = s 'TPCC.Payment'
			InprocServer32 = s '%MODULE%'
			{
				val ThreadingModel = s 'Both'
			}
		}
	}
}
```

tpcc_com_ps.def

```
LIBRARY      "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

EXPORTS
	DllGetClassObject     @1          PRIVATE
	DllCanUnloadNow       @2          PRIVATE
	GetProxyDllInfo       @3          PRIVATE
	DllRegisterServer                 @4          PRIVATE
	DllUnregisterServer               @5          PRIVATE
```

tpcc_com_ps.dsp

```
# Microsoft Developer Studio Project File - Name="tpcc_com_ps" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=tpcc_com_ps - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak" CFG="tpcc_com_ps - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_ps - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "tpcc_com_ps - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tpcc_com_ps - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D _WIN32_WINNT=0x0400 /D
"REGISTER_PROXY_DLL" /FD /c
# SUBTRACT CPP /YX
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib /nologo /entry:"DllMain"
/subsystem:windows /dll /pdb:none /machine:I386 /def:".\src\tpcc_com_ps.def"
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.\bin\tpcc_com_ps.dll
SOURCE=$(InputPath)
```

```
"..\tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
                copy   .\src\tpcc_com_ps.h   ..\tpcc_com_all\src\

# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /Zi /Od /D "WIN32" /D "_DEBUG" /D _WIN32_WINNT=0x0400 /D "REGISTER_PROXY_DLL"
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o NUL /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib /nologo /entry:"DllMain" /dll /debug
/machine:IX86 /def:".\src\tpcc_com_ps.def" /pdbtype:sept
# SUBTRACT LINK32 /pdb:none
# Begin Custom Build - Copyingtpcc_com_ps.h
InputPath=.\bin\tpcc_com_ps.dll
SOURCE=$(InputPath)

"..\tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
                copy   .\src\tpcc_com_ps.h   ..\tpcc_com_all\src\

# End Custom Build

!ENDIF

# Begin Target

# Name "tpcc_com_ps - Win32 Release"
# Name "tpcc_com_ps - Win32 Debug"
# Begin Group "Source"

# PROP Default_Filter ""
# Begin Source File

SOURCE=.\src\dlldata.c
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps.def
# PROP Exclude_From_Build 1
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps.idl

!IF  "$(CFG)" == "tpcc_com_ps - Win32 Release"

# Begin Custom Build
InputPath=.\src\tpcc_com_ps.idl

BuildCmds= \
                midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c" ".\src\tpcc_com_ps.idl"\
  /out ".\src"

".\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)

".\src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)

".\src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)

".\src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)
# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# Begin Custom Build
InputPath=.\src\tpcc_com_ps.idl

BuildCmds= \
                midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c" ".\src\tpcc_com_ps.idl"\
  /out ".\src"

".\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)

".\src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)

".\src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)

".\src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
  $(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
```

```
# Begin Source File

SOURCE=.\src\tpcc_com_ps_i.c
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps_p.c
# End Source File
# End Group
# End Target
# End Project
```

tpcc_com_ps.h

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the interfaces*/


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:17 1999
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )


/* verify that the <rpcndr.h> version is high enough to compile thisfile*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifndef __RPCNDR_H_VERSION__
#error this stub requires an updated version of <rpcndr.h>
#endif // __RPCNDR_H_VERSION__

#ifndef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

#ifndef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif          /* __ITPCC_FWD_DEFINED__ */


/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"

#ifdef __cplusplus
extern "C"{
#endif

void __RPC_FAR * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void __RPC_FAR * );

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

/* interface ITPCC */
/* [unique][helpstring][uuid][object]*/


EXTERN_C const IID IID_ITPCC;

#if defined(__cplusplus) && !defined(CINTERFACE)

    MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall Payment(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall Delivery(
            /* [in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall StockLevel(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall OrderStatus(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn) = 0;

        virtual HRESULT __stdcall CallSetComplete( void) = 0;

    };
```

```
#else           /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

        HRESULT ( STDMETHODCALLTYPE __RPC_FAR *QueryInterface )(
            ITPCC __RPC_FAR * This,
            /* [in] */ REFIID riid,
            /* [iid_is][out] */ void __RPC_FAR *__RPC_FAR *ppvObject);

        ULONG ( STDMETHODCALLTYPE __RPC_FAR *AddRef )(
            ITPCC __RPC_FAR * This);

        ULONG ( STDMETHODCALLTYPE __RPC_FAR *Release )(
            ITPCC __RPC_FAR * This);

        HRESULT ( __stdcall __RPC_FAR *NewOrder )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

        HRESULT ( __stdcall __RPC_FAR *Payment )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

        HRESULT ( __stdcall __RPC_FAR *Delivery )(
            ITPCC __RPC_FAR * This,
            /* [in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

        HRESULT ( __stdcall __RPC_FAR *StockLevel )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

        HRESULT ( __stdcall __RPC_FAR *OrderStatus )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);

        HRESULT ( __stdcall __RPC_FAR *CallSetComplete )(
            ITPCC __RPC_FAR * This);

        END_INTERFACE
    } ITPCCVtbl;

    interface ITPCC
    {
        CONST_VTBL struct ITPCCVtbl __RPC_FAR *lpVtbl;
    };


#ifdef COBJMACROS

#define ITPCC_QueryInterface(This,riid,ppvObject)               \
    (This)->lpVtbl -> QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This)          \
    (This)->lpVtbl -> AddRef(This)

#define ITPCC_Release(This)         \
    (This)->lpVtbl -> Release(This)


#define ITPCC_NewOrder(This,iSize,txn)          \
    (This)->lpVtbl -> NewOrder(This,iSize,txn)

#define ITPCC_Payment(This,iSize,txn)           \
    (This)->lpVtbl -> Payment(This,iSize,txn)

#define ITPCC_Delivery(This,iSize,txn)          \
    (This)->lpVtbl -> Delivery(This,iSize,txn)

#define ITPCC_StockLevel(This,iSize,txn)        \
    (This)->lpVtbl -> StockLevel(This,iSize,txn)

#define ITPCC_OrderStatus(This,iSize,txn)       \
    (This)->lpVtbl -> OrderStatus(This,iSize,txn)

#define ITPCC_CallSetComplete(This)             \
    (This)->lpVtbl -> CallSetComplete(This)

#endif /* COBJMACROS */


#endif          /* C style interface */



HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);


void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR *__RPC_FAR *txn);
```

```
void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);


void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);


void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);


void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);


void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


#endif        /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces*/

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_ps.idl

```
/*            FILE:                ITPCC.IDL
 *                                        Microsoft TPC-C Kit Ver. 4.20.000
 *                                        Copyright Microsoft, 1999
 *        All Rights Reserved
 *
 *                                        not yet audited
 *
 *            PURPOSE:  Defines the interface used by TPCC. This interface can be implemented by C++ components.
 *
 *  Change history:
 *                    4.20.000 - first version
 */

// Forward declare all types defined
//interface ITPCC;
import "oaidl.idl";
import "ocidl.idl";

            [
                    object,
                    uuid(FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),
                    helpstring("ITPCC Interface"),
                    pointer_default(unique)
            ]
            interface ITPCC : IUnknown
            {

    HRESULT _stdcall NewOrder
                                                    (
                                                    [in, out] int* iSize,
                                                    [in, out, size_is( , *iSize)] char** txn
                                                    );

    HRESULT _stdcall Payment
                                                    (
                                                    [in, out] int* iSize,
                                                    [in, out, size_is( , *iSize)] char** txn
```

```
                                                        );

    HRESULT _stdcall Delivery
                                                        (
                                                        [in] int* iSize,
                                                        [in, size_is( , *iSize)] char** txn
                                                        );

    HRESULT _stdcall StockLevel
                                                        (
                                                        [in, out] int* iSize,
                                                        [in, out, size_is( , *iSize)] char** txn
                                                        );

    HRESULT _stdcall OrderStatus
                                                        (
                                                        [in, out] int* iSize,
                                                        [in, out, size_is( , *iSize)] char** txn
                                                        );

    HRESULT _stdcall CallSetComplete
                                                        (
                                                        );

};          // interface ITPCC
```

tpcc_com_ps_i.c

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients*/


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:17 1999
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID, IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* !defined(_M_IA64) && !defined(_M_AXP64)*/


#pragma warning( disable: 4049 )  /* more than 64k source lines */
```

```
/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients*/


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:18 1999
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if defined(_M_IA64) || defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID, IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* defined(_M_IA64) || defined(_M_AXP64)*/
```

tpcc_com_ps_p.c

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the proxy stub code*/


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:17 1999
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AXP64)
#define USE_STUBLESS_PROXY


/* verify that the <rpcproxy.h> version is high enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 440
#endif


#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__
```

```
#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   33
#define PROC_FORMAT_STRING_SIZE   193
#define TRANSMIT_AS_TABLE_SIZE    0
#define WIRE_MARSHAL_TABLE_SIZE   0

typedef struct _MIDL_TYPE_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ TYPE_FORMAT_STRING_SIZE ];
    } MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ PROC_FORMAT_STRING_SIZE ];
    } MIDL_PROC_FORMAT_STRING;


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


/* Object interface: IUnknown, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */


/* Object interface: ITPCC, ver. 0.0,
   GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}}*/


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[]=
    {
    0,
    34,
    68,
    102,
    136,
    170
    };

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
    {
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0
    };

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
    {
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
    };

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *)-1 /* ITPCC::NewOrder */ ,
    (void *)-1 /* ITPCC::Payment */ ,
    (void *)-1 /* ITPCC::Delivery */ ,
    (void *)-1 /* ITPCC::StockLevel */ ,
    (void *)-1 /* ITPCC::OrderStatus */ ,
    (void *)-1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};


static const MIDL_STUB_DESC Object_StubDesc =
    {
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x50200eb, /* MIDL Version 5.2.235 */
```

```
    0,
    0,
    0,  /* notify & notify_flag routine table */
    1,  /* Flags */
    0,  /* Reserved3 */
    0,  /* Reserved4 */
    0   /* Reserved5 */
    };

#pragma data_seg(".rdata")

#if !defined(__RPC_WIN32__)
#error  Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub because it uses these features:
#error   -Oif or -Oicf.
#error However, your C/C++ compilation flags indicate you intend to run this app on earlier systems.
#error This app will die there with the RPC_X_WRONG_STUB_VERSION error.
#endif


static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString=
    {
        0,
        {

            /* Procedure NewOrder */

                                        0x33,               /* FC_AUTO_HANDLE */
                                        0x6c,               /* Old Flags:  object, Oi2 */
/*  2 */    NdrFcLong( 0x0 ),           /* 0 */
/*  6 */    NdrFcShort( 0x3 ),          /* 3 */
#ifndef _ALPHA_
/*  8 */    NdrFcShort( 0x10 ),         /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                        NdrFcShort( 0x20 ),         /* Alpha Stack size/offset = 32 */
#endif
/* 10 */    NdrFcShort( 0x8 ),          /* 8 */
/* 12 */    NdrFcShort( 0x10 ),         /* 16 */
/* 14 */    0x7,                        /* Oi2 Flags:  srv must size, clt must size, has return, */
                                        0x3,                /* 3 */

            /* Parameter iSize */

/* 16 */    NdrFcShort( 0x158 ),        /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 18 */    NdrFcShort( 0x4 ),          /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                        NdrFcShort( 0x8 ),          /* Alpha Stack size/offset = 8 */
#endif
/* 20 */    0x8,                        /* FC_LONG */
                                        0x0,                /* 0 */

            /* Parameter txn */

/* 22 */    NdrFcShort( 0x201b ),       /* Flags:  must size, must free, in, out, srv alloc size=8 */
#ifndef _ALPHA_
/* 24 */    NdrFcShort( 0x8 ),          /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                        NdrFcShort( 0x10 ),         /* Alpha Stack size/offset = 16 */
#endif
/* 26 */    NdrFcShort( 0x6 ),          /* Type Offset=6 */

            /* Return value */

/* 28 */    NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 30 */    NdrFcShort( 0xc ),          /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                        NdrFcShort( 0x18 ),         /* Alpha Stack size/offset = 24 */
#endif
/* 32 */    0x8,                        /* FC_LONG */
                                        0x0,                /* 0 */

            /* Procedure Payment */

/* 34 */    0x33,                       /* FC_AUTO_HANDLE */
                                        0x6c,               /* Old Flags:  object, Oi2 */
/* 36 */    NdrFcLong( 0x0 ),           /* 0 */
/* 40 */    NdrFcShort( 0x4 ),          /* 4 */
#ifndef _ALPHA_
/* 42 */    NdrFcShort( 0x10 ),         /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                        NdrFcShort( 0x20 ),         /* Alpha Stack size/offset = 32 */
#endif
/* 44 */    NdrFcShort( 0x8 ),          /* 8 */
/* 46 */    NdrFcShort( 0x10 ),         /* 16 */
/* 48 */    0x7,                        /* Oi2 Flags:  srv must size, clt must size, has return, */
                                        0x3,                /* 3 */

            /* Parameter iSize */

/* 50 */    NdrFcShort( 0x158 ),        /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 52 */    NdrFcShort( 0x4 ),          /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                        NdrFcShort( 0x8 ),          /* Alpha Stack size/offset = 8 */
#endif
/* 54 */    0x8,                        /* FC_LONG */
                                        0x0,                /* 0 */

            /* Parameter txn */

/* 56 */    NdrFcShort( 0x201b ),       /* Flags:  must size, must free, in, out, srv alloc size=8 */
#ifndef _ALPHA_
/* 58 */    NdrFcShort( 0x8 ),          /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                        NdrFcShort( 0x10 ),         /* Alpha Stack size/offset = 16 */
#endif
/* 60 */    NdrFcShort( 0x6 ),          /* Type Offset=6 */
```

```
            /* Return value */

/* 62 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 64 */    NdrFcShort( 0xc ),      /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                    NdrFcShort( 0x18 ),     /* Alpha Stack size/offset = 24 */
#endif
/* 66 */    0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Procedure Delivery */

/* 68 */    0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /* Old Flags:  object, Oi2 */
/* 70 */    NdrFcLong( 0x0 ),       /* 0 */
/* 74 */    NdrFcShort( 0x5 ),      /* 5 */
#ifndef _ALPHA_
/* 76 */    NdrFcShort( 0x10 ),     /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                    NdrFcShort( 0x20 ),     /* Alpha Stack size/offset = 32 */
#endif
/* 78 */    NdrFcShort( 0x8 ),      /* 8 */
/* 80 */    NdrFcShort( 0x8 ),      /* 8 */
/* 82 */    0x6,                    /* Oi2 Flags:  clt must size, has return, */
                                    0x3,                    /* 3 */

            /* Parameter iSize */

/* 84 */    NdrFcShort( 0x148 ),    /* Flags:  in, base type, simple ref, */
#ifndef _ALPHA_
/* 86 */    NdrFcShort( 0x4 ),      /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                    NdrFcShort( 0x8 ),      /* Alpha Stack size/offset = 8 */
#endif
/* 88 */    0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Parameter txn */

/* 90 */    NdrFcShort( 0x200b ),   /* Flags:  must size, must free, in, srv alloc size=8 */
#ifndef _ALPHA_
/* 92 */    NdrFcShort( 0x8 ),      /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                    NdrFcShort( 0x10 ),     /* Alpha Stack size/offset = 16 */
#endif
/* 94 */    NdrFcShort( 0x18 ),     /* Type Offset=24 */

            /* Return value */

/* 96 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 98 */    NdrFcShort( 0xc ),      /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                    NdrFcShort( 0x18 ),     /* Alpha Stack size/offset = 24 */
#endif
/* 100 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Procedure StockLevel */

/* 102 */   0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /* Old Flags:  object, Oi2 */
/* 104 */   NdrFcLong( 0x0 ),       /* 0 */
/* 108 */   NdrFcShort( 0x6 ),      /* 6 */
#ifndef _ALPHA_
/* 110 */   NdrFcShort( 0x10 ),     /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                    NdrFcShort( 0x20 ),     /* Alpha Stack size/offset = 32 */
#endif
/* 112 */   NdrFcShort( 0x8 ),      /* 8 */
/* 114 */   NdrFcShort( 0x10 ),     /* 16 */
/* 116 */   0x7,                    /* Oi2 Flags:  srv must size, clt must size, has return, */
                                    0x3,                    /* 3 */

            /* Parameter iSize */

/* 118 */   NdrFcShort( 0x158 ),    /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 120 */   NdrFcShort( 0x4 ),      /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                    NdrFcShort( 0x8 ),      /* Alpha Stack size/offset = 8 */
#endif
/* 122 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Parameter txn */

/* 124 */   NdrFcShort( 0x201b ),   /* Flags:  must size, must free, in, out, srv alloc size=8 */
#ifndef _ALPHA_
/* 126 */   NdrFcShort( 0x8 ),      /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                    NdrFcShort( 0x10 ),     /* Alpha Stack size/offset = 16 */
#endif
/* 128 */   NdrFcShort( 0x6 ),      /* Type Offset=6 */

            /* Return value */

/* 130 */   NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 132 */   NdrFcShort( 0xc ),      /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                    NdrFcShort( 0x18 ),     /* Alpha Stack size/offset = 24 */
#endif
/* 134 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0 */

            /* Procedure OrderStatus */

/* 136 */   0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /* Old Flags:  object, Oi2 */
```

```
/* 138 */      NdrFcLong( 0x0 ),        /* 0 */
/* 142 */      NdrFcShort( 0x7 ),       /* 7 */
#ifndef _ALPHA_
/* 144 */      NdrFcShort( 0x10 ),      /* x86, MIPS, PPC Stack size/offset = 16 */
#else
                                        NdrFcShort( 0x20 ),      /* Alpha Stack size/offset = 32 */
#endif
/* 146 */      NdrFcShort( 0x8 ),       /* 8 */
/* 148 */      NdrFcShort( 0x10 ),      /* 16 */
/* 150 */      0x7,                     /* Oi2 Flags:  srv must size, clt must size, has return, */
                                        0x3,                     /* 3 */

            /* Parameter iSize */

/* 152 */      NdrFcShort( 0x158 ),     /* Flags:  in, out, base type, simple ref, */
#ifndef _ALPHA_
/* 154 */      NdrFcShort( 0x4 ),       /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                        NdrFcShort( 0x8 ),       /* Alpha Stack size/offset = 8 */
#endif
/* 156 */      0x8,                     /* FC_LONG */
                                        0x0,                     /* 0 */

            /* Parameter txn */

/* 158 */      NdrFcShort( 0x201b ),    /* Flags:  must size, must free, in, out, srv alloc size=8 */
#ifndef _ALPHA_
/* 160 */      NdrFcShort( 0x8 ),       /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                        NdrFcShort( 0x10 ),      /* Alpha Stack size/offset = 16 */
#endif
/* 162 */      NdrFcShort( 0x6 ),       /* Type Offset=6 */

            /* Return value */

/* 164 */      NdrFcShort( 0x70 ),      /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 166 */      NdrFcShort( 0xc ),       /* x86, MIPS, PPC Stack size/offset = 12 */
#else
                                        NdrFcShort( 0x18 ),      /* Alpha Stack size/offset = 24 */
#endif
/* 168 */      0x8,                     /* FC_LONG */
                                        0x0,                     /* 0 */

            /* Procedure CallSetComplete */

/* 170 */      0x33,                    /* FC_AUTO_HANDLE */
                                        0x6c,                    /* Old Flags:  object, Oi2 */
/* 172 */      NdrFcLong( 0x0 ),        /* 0 */
/* 176 */      NdrFcShort( 0x8 ),       /* 8 */
#ifndef _ALPHA_
/* 178 */      NdrFcShort( 0x8 ),       /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                                        NdrFcShort( 0x10 ),      /* Alpha Stack size/offset = 16 */
#endif
/* 180 */      NdrFcShort( 0x0 ),       /* 0 */
/* 182 */      NdrFcShort( 0x8 ),       /* 8 */
/* 184 */      0x4,                     /* Oi2 Flags:  has return, */
                                        0x1,                     /* 1 */

            /* Return value */

/* 186 */      NdrFcShort( 0x70 ),      /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 188 */      NdrFcShort( 0x4 ),       /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                        NdrFcShort( 0x8 ),       /* Alpha Stack size/offset = 8 */
#endif
/* 190 */      0x8,                     /* FC_LONG */
                                        0x0,                     /* 0 */

                                        0x0
        }
    };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString=
    {
        0,
        {
                                        NdrFcShort( 0x0 ),       /* 0 */
/*  2 */
                                        0x11, 0x8,   /* FC_RP [simple_pointer] */
/*  4 */       0x8,                     /* FC_LONG */
                                        0x5c,                    /* FC_PAD */
/*  6 */
                                        0x11, 0x14,  /* FC_RP [alloced_on_stack] [pointer_deref] */
/*  8 */       NdrFcShort( 0x2 ),       /* Offset= 2 (10) */
/* 10 */
                                        0x13, 0x0,   /* FC_OP */
/* 12 */       NdrFcShort( 0x2 ),       /* Offset= 2 (14) */
/* 14 */
                                        0x1b,                    /* FC_CARRAY */
                                        0x0,                     /* 0 */
/* 16 */       NdrFcShort( 0x1 ),       /* 1 */
/* 18 */       0x28,                    /* Corr desc:  parameter, FC_LONG */
                                        0x54,                    /* FC_DEREFERENCE */
#ifndef _ALPHA_
/* 20 */       NdrFcShort( 0x4 ),       /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                                        NdrFcShort( 0x8 ),       /* Alpha Stack size/offset = 8 */
#endif
/* 22 */       0x2,                     /* FC_CHAR */
                                        0x5b,                    /* FC_END */
/* 24 */
                                        0x11, 0x14,  /* FC_RP [alloced_on_stack] [pointer_deref] */
/* 26 */       NdrFcShort( 0x2 ),       /* Offset= 2 (28) */
/* 28 */
                                        0x12, 0x0,   /* FC_UP */
/* 30 */       NdrFcShort( 0xfffffff0), /* Offset= -16 (14) */

                                        0x0
        }
```

```
    };

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};


#define _tpcc_com_ps_CHECK_IID(n)                IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{

    if(!_tpcc_com_ps_CHECK_IID(0))
        {
        *pIndex = 0;
        return 1;
        }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo=
{
    (PCInterfaceProxyVtblList *) & _tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) & _tpcc_com_ps_StubVtblList,
    (const PCInterfaceName * ) & _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0  /* Filler3 */
};


#endif /* !defined(_M_IA64) && !defined(_M_AXP64)*/


#pragma warning( disable: 4049 )  /* more than 64k source lines */
```

```
/* this ALWAYS GENERATED file contains the proxy stub code*/


 /* File created by MIDL compiler version 5.02.0235 */
/* at Fri Aug 13 18:56:18 1999
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if defined(_M_IA64) || defined(_M_AXP64)
#define USE_STUBLESS_PROXY


/* verify that the <rpcproxy.h> version is high enough to compile thisfile*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 440
#endif


#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__


#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   33
#define PROC_FORMAT_STRING_SIZE   193
#define TRANSMIT_AS_TABLE_SIZE    0
#define WIRE_MARSHAL_TABLE_SIZE   0

typedef struct _MIDL_TYPE_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ TYPE_FORMAT_STRING_SIZE ];
    } MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ PROC_FORMAT_STRING_SIZE ];
    } MIDL_PROC_FORMAT_STRING;


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


/* Object interface: IUnknown, ver. 0.0,
```

```
   GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */


/* Object interface: ITPCC, ver. 0.0,
   GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}}*/


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[]=
    {
    0,
    34,
    68,
    102,
    136,
    170
    };

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
    {
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0
    };

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo=
    {
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
    };

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *)-1 /* ITPCC::NewOrder */ ,
    (void *)-1 /* ITPCC::Payment */ ,
    (void *)-1 /* ITPCC::Delivery */ ,
    (void *)-1 /* ITPCC::StockLevel */ ,
    (void *)-1 /* ITPCC::OrderStatus */ ,
    (void *)-1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};


static const MIDL_STUB_DESC Object_StubDesc =
    {
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x50200eb, /* MIDL Version 5.2.235 */
    0,
    0,
    0,  /* notify & notify_flag routine table */
    1,  /* Flags */
    0,  /* Reserved3 */
    0,  /* Reserved4 */
    0   /* Reserved5 */
    };

#pragma data_seg(".rdata")

#if !defined(__RPC_WIN64__)
#error  Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString=
    {
        0,
        {

            /* Procedure NewOrder */

                                        0x33,           /* FC_AUTO_HANDLE */
                                        0x6c,           /* Old Flags:  object, Oi2 */
/*  2 */    NdrFcLong( 0x0 ),       /* 0 */
/*  6 */    NdrFcShort( 0x3 ),      /* 3 */
/*  8 */    NdrFcShort( 0x20 ),     /* ia64, axp64 Stack size/offset = 32 */
```

```
/* 10 */    NdrFcShort( 0x8 ),      /* 8 */
/* 12 */    NdrFcShort( 0x10 ),     /* 16 */
/* 14 */    0x7,                    /* Oi2 Flags:  srv must size, clt must size, has return, */
            0x3,                    /* 3 */

            /* Parameter iSize */

/* 16 */    NdrFcShort( 0x158 ),    /* Flags:  in, out, base type, simple ref, */
/* 18 */    NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 20 */    0x8,                    /* FC_LONG */
            0x0,                    /* 0 */

            /* Parameter txn */

/* 22 */    NdrFcShort( 0x201b ),   /* Flags:  must size, must free, in, out, srv alloc size=8 */
/* 24 */    NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 26 */    NdrFcShort( 0x6 ),      /* Type Offset=6 */

            /* Return value */

/* 28 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 30 */    NdrFcShort( 0x18 ),     /* ia64, axp64 Stack size/offset = 24 */
/* 32 */    0x8,                    /* FC_LONG */
            0x0,                    /* 0 */

            /* Procedure Payment */

/* 34 */    0x33,                   /* FC_AUTO_HANDLE */
            0x6c,                   /* Old Flags:  object, Oi2 */
/* 36 */    NdrFcLong( 0x0 ),       /* 0 */
/* 40 */    NdrFcShort( 0x4 ),      /* 4 */
/* 42 */    NdrFcShort( 0x20 ),     /* ia64, axp64 Stack size/offset = 32 */
/* 44 */    NdrFcShort( 0x8 ),      /* 8 */
/* 46 */    NdrFcShort( 0x10 ),     /* 16 */
/* 48 */    0x7,                    /* Oi2 Flags:  srv must size, clt must size, has return, */
            0x3,                    /* 3 */

            /* Parameter iSize */

/* 50 */    NdrFcShort( 0x158 ),    /* Flags:  in, out, base type, simple ref, */
/* 52 */    NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 54 */    0x8,                    /* FC_LONG */
            0x0,                    /* 0 */

            /* Parameter txn */

/* 56 */    NdrFcShort( 0x201b ),   /* Flags:  must size, must free, in, out, srv alloc size=8 */
/* 58 */    NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 60 */    NdrFcShort( 0x6 ),      /* Type Offset=6 */

            /* Return value */

/* 62 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 64 */    NdrFcShort( 0x18 ),     /* ia64, axp64 Stack size/offset = 24 */
/* 66 */    0x8,                    /* FC_LONG */
            0x0,                    /* 0 */

            /* Procedure Delivery */

/* 68 */    0x33,                   /* FC_AUTO_HANDLE */
            0x6c,                   /* Old Flags:  object, Oi2 */
/* 70 */    NdrFcLong( 0x0 ),       /* 0 */
/* 74 */    NdrFcShort( 0x5 ),      /* 5 */
/* 76 */    NdrFcShort( 0x20 ),     /* ia64, axp64 Stack size/offset = 32 */
/* 78 */    NdrFcShort( 0x8 ),      /* 8 */
/* 80 */    NdrFcShort( 0x8 ),      /* 8 */
/* 82 */    0x6,                    /* Oi2 Flags:  clt must size, has return, */
            0x3,                    /* 3 */

            /* Parameter iSize */

/* 84 */    NdrFcShort( 0x148 ),    /* Flags:  in, base type, simple ref, */
/* 86 */    NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 88 */    0x8,                    /* FC_LONG */
            0x0,                    /* 0 */

            /* Parameter txn */

/* 90 */    NdrFcShort( 0x200b ),   /* Flags:  must size, must free, in, srv alloc size=8 */
/* 92 */    NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 94 */    NdrFcShort( 0x18 ),     /* Type Offset=24 */

            /* Return value */

/* 96 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 98 */    NdrFcShort( 0x18 ),     /* ia64, axp64 Stack size/offset = 24 */
/* 100 */   0x8,                    /* FC_LONG */
            0x0,                    /* 0 */

            /* Procedure StockLevel */

/* 102 */   0x33,                   /* FC_AUTO_HANDLE */
            0x6c,                   /* Old Flags:  object, Oi2 */
/* 104 */   NdrFcLong( 0x0 ),       /* 0 */
/* 108 */   NdrFcShort( 0x6 ),      /* 6 */
/* 110 */   NdrFcShort( 0x20 ),     /* ia64, axp64 Stack size/offset = 32 */
/* 112 */   NdrFcShort( 0x8 ),      /* 8 */
/* 114 */   NdrFcShort( 0x10 ),     /* 16 */
/* 116 */   0x7,                    /* Oi2 Flags:  srv must size, clt must size, has return, */
            0x3,                    /* 3 */

            /* Parameter iSize */

/* 118 */   NdrFcShort( 0x158 ),    /* Flags:  in, out, base type, simple ref, */
/* 120 */   NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 122 */   0x8,                    /* FC_LONG */
            0x0,                    /* 0 */

            /* Parameter txn */

/* 124 */   NdrFcShort( 0x201b ),   /* Flags:  must size, must free, in, out, srv alloc size=8 */
/* 126 */   NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
```

```
/* 128 */    NdrFcShort( 0x6 ),      /* Type Offset=6 */

             /* Return value */

/* 130 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 132 */    NdrFcShort( 0x18 ),     /* ia64, axp64 Stack size/offset = 24 */
/* 134 */    0x8,                    /* FC_LONG */
                                     0x0,                    /* 0 */

             /* Procedure OrderStatus */

/* 136 */    0x33,                   /* FC_AUTO_HANDLE */
                                     0x6c,                   /* Old Flags:  object, Oi2 */
/* 138 */    NdrFcLong( 0x0 ),       /* 0 */
/* 142 */    NdrFcShort( 0x7 ),      /* 7 */
/* 144 */    NdrFcShort( 0x20 ),     /* ia64, axp64 Stack size/offset = 32 */
/* 146 */    NdrFcShort( 0x8 ),      /* 8 */
/* 148 */    NdrFcShort( 0x10 ),     /* 16 */
/* 150 */    0x7,                    /* Oi2 Flags:  srv must size, clt must size, has return, */
                                     0x3,                    /* 3 */

             /* Parameter iSize */

/* 152 */    NdrFcShort( 0x158 ),    /* Flags:  in, out, base type, simple ref, */
/* 154 */    NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 156 */    0x8,                    /* FC_LONG */
                                     0x0,                    /* 0 */

             /* Parameter txn */

/* 158 */    NdrFcShort( 0x201b ),   /* Flags:  must size, must free, in, out, srv alloc size=8 */
/* 160 */    NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 162 */    NdrFcShort( 0x6 ),      /* Type Offset=6 */

             /* Return value */

/* 164 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 166 */    NdrFcShort( 0x18 ),     /* ia64, axp64 Stack size/offset = 24 */
/* 168 */    0x8,                    /* FC_LONG */
                                     0x0,                    /* 0 */

             /* Procedure CallSetComplete */

/* 170 */    0x33,                   /* FC_AUTO_HANDLE */
                                     0x6c,                   /* Old Flags:  object, Oi2 */
/* 172 */    NdrFcLong( 0x0 ),       /* 0 */
/* 176 */    NdrFcShort( 0x8 ),      /* 8 */
/* 178 */    NdrFcShort( 0x10 ),     /* ia64, axp64 Stack size/offset = 16 */
/* 180 */    NdrFcShort( 0x0 ),      /* 0 */
/* 182 */    NdrFcShort( 0x8 ),      /* 8 */
/* 184 */    0x4,                    /* Oi2 Flags:  has return, */
                                     0x1,                    /* 1 */

             /* Return value */

/* 186 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base type, */
/* 188 */    NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 190 */    0x8,                    /* FC_LONG */
                                     0x0,                    /* 0 */

                                     0x0
    }
  };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString=
  {
    0,
    {
                                     NdrFcShort( 0x0 ),      /* 0 */
/*  2 */
                                     0x11, 0x8,  /* FC_RP [simple_pointer] */
/*  4 */    0x8,                    /* FC_LONG */
                                     0x5c,                   /* FC_PAD */
/*  6 */
                                     0x11, 0x14,  /* FC_RP [alloced_on_stack] [pointer_deref] */
/*  8 */    NdrFcShort( 0x2 ),      /* Offset= 2 (10) */
/* 10 */
                                     0x13, 0x0,  /* FC_OP */
/* 12 */    NdrFcShort( 0x2 ),      /* Offset= 2 (14) */
/* 14 */
                                     0x1b,                   /* FC_CARRAY */
                                     0x0,                    /* 0 */
/* 16 */    NdrFcShort( 0x1 ),      /* 1 */
/* 18 */    0x28,                   /* Corr desc:  parameter, FC_LONG */
                                     0x54,                   /* FC_DEREFERENCE */
/* 20 */    NdrFcShort( 0x8 ),      /* ia64, axp64 Stack size/offset = 8 */
/* 22 */    0x2,                    /* FC_CHAR */
                                     0x5b,                   /* FC_END */
/* 24 */
                                     0x11, 0x14,  /* FC_RP [alloced_on_stack] [pointer_deref] */
/* 26 */    NdrFcShort( 0x2 ),      /* Offset= 2 (28) */
/* 28 */
                                     0x12, 0x0,  /* FC_UP */
/* 30 */    NdrFcShort( 0xffffffd0),/* Offset= -16 (14) */

                                     0x0
    }
  };

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[]=
{
  ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
  0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[]=
{
  ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
  0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[]=
```

```
{
   "ITPCC",
   0
};


#define _tpcc_com_ps_CHECK_IID(n)              IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{

   if(!_tpcc_com_ps_CHECK_IID(0))
      {
      *pIndex = 0;
      return 1;
      }

   return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo=
{
   (PCInterfaceProxyVtblList *) & _tpcc_com_ps_ProxyVtblList,
   (PCInterfaceStubVtblList *) & _tpcc_com_ps_StubVtblList,
   (const PCInterfaceName * ) & _tpcc_com_ps_InterfaceNamesList,
   0, // no delegation
   & _tpcc_com_ps_IID_Lookup,
   1,
   2,
   0, /* table of [async_uuid] interfaces */
   0, /* Filler1 */
   0, /* Filler2 */
   0  /* Filler3 */
};


#endif /* defined(_M_IA64) || defined(_M_AXP64)*/
```

tpcc_com_resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc_com_all.rc
//
#define IDS_PROJNAME                100
#define IDR_TPCC                    101
#define IDR_NEWORDER                102
#define IDR_ORDERSTATUS             103
#define IDR_PAYMENT                 104
#define IDR_STOCKLEVEL              105

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        202
#define _APS_NEXT_COMMAND_VALUE         32768
#define _APS_NEXT_CONTROL_VALUE         201
#define _APS_NEXT_SYMED_VALUE           106
#endif
#endif
```

tpcc_com_sl.rgs

```
HKCR
{
        TPCC.StockLevel.1 = s 'StockLevel Class'
        {
                CLSID = s '{2668369E-A50D-11D2-BA4E-00C04FBFE08B}'
        }
        TPCC.StockLevel = s 'StockLevel Class'
        {
                CurVer = s 'TPCC.StockLevel.1'
        }
        NoRemove CLSID
        {
                ForceRemove {2668369E-A50D-11D2-BA4E-00C04FBFE08B} = s 'StockLevel Class'
                {
                        ProgID = s 'TPCC.StockLevel.1'
                        VersionIndependentProgID = s 'TPCC.StockLevel'
                        InprocServer32 = s '%MODULE%'
                        {
                                val ThreadingModel = s 'Both'
                        }
                }
        }
}
```

tpcc_dblib.cpp

```
/*      FILE:                   TPCC_DBLIB.CPP
 *                                      Microsoft TPC-C Kit Ver. 4.20.000
 *                                      Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                                      Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 *      PURPOSE:  Implements dblib calls for TPC-C txns.
 *      Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 *                      4.10.001 - not deleting error class in catch handler on deadlock retry;
```

```
 *                                                      not a functional bug, but a memory leak
 *                                                    - had to tweak some declarations to compile with latest SDK; no
functional change
 */

#include <windows.h>
#include <stdio.h>
#include <assert.h>

#define DBNTWIN32
#include <sqlfront.h>
#include <sqldb.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl  __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_dblib.h"

#define       DEFCLPACKSIZE                                   4096

// version string; must match return value fromtpcc_version stored proc
const char    sVersion[] = "4.10.000";

const                               iMaxRetries = 10;                       // how many retries on deadlock
static        long          iConnectionCount = 0;  // number of current dblib connections

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
              switch( ul_reason_for_call )
              {
                            case DLL_PROCESS_ATTACH:
                                          DisableThreadLibraryCalls(hModule);
                                          dbinit();                       // initialize dblib
                                          break;

                            case DLL_PROCESS_DETACH:
                                          dbexit();                       // close all dblib structures/connections
                                          break;

                            default:
                                          /* nothing */;
              }
              return TRUE;
}


int err_handler(DBPROCESS *dbproc, int severity, int dberr, int oserr, LPCSTR dberrstr, LPCSTR oserrstr)
{
              CTPCC_DBLIB                                     *pConn;

              assert(dbproc != NULL);
              pConn = (CTPCC_DBLIB*)dbgetuserdata(dbproc);

              if (pConn != NULL)
              {
                            pConn->SetDbLibError( severity, dberr, oserr, dberrstr, oserrstr );
              }
              return INT_CANCEL;
}

/* FUNCTION: int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity, char *msgtext)
 *
 * PURPOSE:              This function handles DB-Library SQL Server error messages
 *
 * ARGUMENTS:    DBPROCESS                       *dbproc                                 DBPROCESS id
pointer
 *                                                      DBINT                                   msgno
message number
 *                                                      int                                             msgstate
message state
 *                                                      int                                             severity
message severity
 *                                                      char                                    *msgtext
printable message description
 *
 * RETURNS:                    int                                                     INT_CONTINUE
continue if error is SQLETIME else INT_CANCEL action
 *                                                                                                      INT_CANCEL
              cancel operation
 *
 * COMMENTS:             This function also sets the dead lock dbproc variable if necessary.
 *
 */

// typedef INT (SQLAPI *DBMSGHANDLE_PROC)(PDBPROCESS, DBINT, INT, INT, LPCSTR, LPCSTR, LPCSTR,
DBUSMALLINT);

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity,
                                                      LPCSTR msgtext, LPCSTR srvname, LPCSTR procname,
DBUSMALLINT line)
{
              CTPCC_DBLIB                                     *pConn;

              assert(dbproc != NULL);
              pConn = (CTPCC_DBLIB*)dbgetuserdata(dbproc);

              if (pConn != NULL)
              {
                            pConn->SetSqlError( msgno, msgstate, severity, msgtext );
              }

              return 0;
}

/* FUNCTION: void UtilStrCpy(char * pDest, char * pSrc, int n)
```

```
 *
 * PURPOSE:             This function copies n characters from stringpSrc to pDst and places a
 *                              null character at the end of the destination string.
 *
 * ARGUMENTS:           char                                    *pDest          destination string pointer
 *                                      char                                            *pSrc           source string pointer
 *                                      int                                                             n
number of characters to copy
 *
 * RETURNS:                     None
 *
 * COMMENTS:            Unlike strncpy this function ensures that the result string is
 *                                      always null terminated.
 *
 */

inline static void UtilStrCpy(char * pDest, const BYTE * pSrc, int n)
{
   strncpy(pDest, (char *)pSrc, n);
   pDest[n] = '\0';

        return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
 *
 */

char* CTPCC_DBLIB_ERR::ErrorText(void)
{
        int i;

        static SERRORMSG errorMsgs[] =
        {
                { ERR_WRONG_SP_VERSION,         "Wrong version of stored procs on database
server"  },
                { ERR_INVALID_CUST,                             "Invalid Customer id,name."
                },
                { ERR_NO_SUCH_ORDER,            "No orders found for customer."
                        },
                { 0,                                                    ""

        }
        };

        static char szNotFound[] = "Unknown error number.";

        for(i=0; errorMsgs[i].szMsg[0]; i++)
        {
                if ( m_errno == errorMsgs[i].iError )
                        break;
        }
        if ( !errorMsgs[i].szMsg[0] )
                return szNotFound;
        else
                return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_DBLIB* CTPCC_DBLIB_new(
        LPCSTR szServer,                // name of SQL server
        LPCSTR szUser,                          // user name for login
        LPCSTR szPassword,              // password for login
        LPCSTR szHost,                          // workstation name; shows up in sp_who; max 30 chars,
only first 10 kept by SQL Server
        LPCSTR szDatabase )             // name of database to use
{
        return new CTPCC_DBLIB( szServer, szUser, szPassword, szHost, szDatabase );
}


CTPCC_DBLIB::CTPCC_DBLIB(
        LPCSTR szServer,                // name of SQL server
        LPCSTR szUser,                          // user name for login
        LPCSTR szPassword,              // password for login
        LPCSTR szHost,                          // workstation name; shows up in sp_who; max 30 chars,
only first 10 kept by SQL Server
        LPCSTR szDatabase )             // name of database to use
{
        LOGINREC *login;
        const BYTE *pData;

        // initialization
        m_dbproc = NULL;
        m_DbLibErr = (CDBLIBERR*)NULL;
        m_SqlErr = (CSQLERR*)NULL;

        m_MaxRetries = 10;              // how many retries on deadlock

        // increase max number of connections if getting close
        if ( dbgetmaxprocs() < (iConnectionCount+5) )
        {
                if ( dbsetmaxprocs(iConnectionCount+10) == FAIL )
                        ThrowError(CDBLIBERR::eDbSetMaxProcs);
        }

        // allocate a login structure
        login = dblogin();
        if (login == NULL)
                ThrowError(CDBLIBERR::eLogin);
        InterlockedIncrement( &iConnectionCount );

        // register error and message handler functions
        if (dbprocerrhandle(login, err_handler) == NULL)
                ThrowError(CDBLIBERR::eDbProcHandler);

        if (dbprocmsghandle(login, msg_handler) == NULL)
                ThrowError(CDBLIBERR::eDbProcHandler);


        DBSETLUSER(login, szUser);
        DBSETLPWD(login, szPassword);
```

```
DBSETLHOST(login, szHost);
DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);
DBSETLVERSION(login, DBVER60);                          // use dblib ver 6.0 client behavior

// set time to wait for login
if (dbsetlogintime(60) == FAIL)
        ThrowError(CDBLIBERR::eDbSet);

// set time to wait for statement execution
if (dbsettime(180) == FAIL)
        ThrowError(CDBLIBERR::eDbSet);

m_dbproc = dbopen(login, szServer);

// deallocate login structure before checking for success
dbfreelogin( login );

if (m_dbproc == NULL)
        ThrowError(CDBLIBERR::eDbOpen);

// save address of class instance so that the message and error handler
// can get to data.
dbsetuserdata(m_dbproc, (LPVOID)this);

// Use the the right database
if (dbuse(m_dbproc, szDatabase) == FAIL)
        ThrowError(CDBLIBERR::eDbUse);

dbcmd(m_dbproc, "set nocount on ");                     // do not return row counts
dbcmd(m_dbproc, "set XACT_ABORT ON");                   // rollback transaction on abort

if (dbsqlexec(m_dbproc) == FAIL)
        ThrowError(CDBLIBERR::eDbSqlExec);

DiscardNextResults(2);

// verify that version of stored procs on server is correct
dbrpcinit(m_dbproc, "tpcc_version", 0);

if (dbrpcexec(m_dbproc) == FAIL)
        ThrowError(CDBLIBERR::eDbRpcExec);

if (dbresults(m_dbproc) != SUCCEED)
        ThrowError(CDBLIBERR::eDbResults);

if (dbnextrow(m_dbproc) != REG_ROW)
        ThrowError(CDBLIBERR::eDbNextRow);

char szSrvVersion[16];
pData=dbdata(m_dbproc, 1);
if (pData)
        UtilStrCpy(szSrvVersion, pData, dbdatlen(m_dbproc, 1));
else
        szSrvVersion[0]=0;
if (strcmp(szSrvVersion,sVersion))
        throw new CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_WRONG_SP_VERSION );
        DiscardNextRows(0);
        DiscardNextResults(0);
}


CTPCC_DBLIB::~CTPCC_DBLIB( void )
{
        // close db connection and deallocate resources
        dbclose(m_dbproc);
        InterlockedDecrement( &iConnectionCount );
        if (m_DbLibErr != NULL)
                delete m_DbLibErr;
        if (m_SqlErr != NULL)
                delete m_SqlErr;
}

void CTPCC_DBLIB::SetDbLibError(int severity, int dberr, int oserr, LPCSTR dberrstr, LPCSTR oserrstr)
{
        delete m_DbLibErr;
        m_DbLibErr = new CDBLIBERR(CDBLIBERR::eUnknown, severity, dberr, oserr);

        if (dberrstr != NULL)
        {
                m_DbLibErr->m_dberrstr = new char[ strlen(dberrstr)+1 ];
                strcpy( m_DbLibErr->m_dberrstr, dberrstr );
        }

        if (oserrstr != NULL)
        {
                m_DbLibErr->m_oserrstr = new char[ strlen(oserrstr)+1 ];
                strcpy( m_DbLibErr->m_oserrstr, oserrstr );
        }

}

void CTPCC_DBLIB::SetSqlError( int /*DBINT*/ msgno, int msgstate, int severity, LPCSTR msgtext )
{
        if (m_SqlErr == NULL)
                m_SqlErr = new CSQLERR();

        m_SqlErr->m_msgno = msgno;
        m_SqlErr->m_msgstate = msgstate;
        m_SqlErr->m_severity = severity;

        delete [] m_SqlErr->m_msgtext;
        if (msgtext != NULL)
        {
                m_SqlErr->m_msgtext = new char[ strlen(msgtext)+1 ];
                strcpy( m_SqlErr->m_msgtext, msgtext );
        }
}

void CTPCC_DBLIB::ThrowError( CDBLIBERR::ACTION eAction )
{
```

```
		// discard anything still in return buffer
		DiscardNextRows(-1);
		DiscardNextResults(-1);

		// check for SQL Server error first;  if yes, throw it and ignore any DBLib error.
		if (m_SqlErr != NULL)
		{
			CSQLERR		*pSqlErr;
			pSqlErr = m_SqlErr;
			m_SqlErr = NULL;	// clear our pointer to instance; catch handler will delete
			throw pSqlErr;
		}

		CDBLIBERR		*pDbLibErr;
		if (m_DbLibErr == NULL)
			// this case isn't expected to happen, since it means that an error was returned
			// but the error handlers were not called.
			pDbLibErr = new CDBLIBERR(eAction);
		else
		{
			pDbLibErr = m_DbLibErr;
			pDbLibErr->m_eAction = eAction;
			m_DbLibErr = NULL;			// clear our pointer to instance; catch handler will delete
		}

		throw pDbLibErr;
}

// Read and discard rows until no more.  Throw an exception if number of rows read doesn't
// match number of rows expected.  The row count will be ignored if the expected count value
// passed in is negative.  A typical use of this routine is to verify that there are no more
// rows to be read.
void CTPCC_DBLIB::DiscardNextRows(int iExpectedCount)
{
		int			iRowsRead = 0;
		RETCODE  rc;

		while (TRUE)
		{
			rc = dbnextrow(m_dbproc);
			if (rc == NO_MORE_ROWS)
				break;
			if (rc == FAIL)
			{
				if (iExpectedCount >= 0)
					ThrowError(CDBLIBERR::eDbNextRow);
				else
					break;
			}
			iRowsRead++;
		}

		if ((iExpectedCount >= 0) &&
			(iExpectedCount != iRowsRead))
			ThrowError(CDBLIBERR::eWrongRowCount);
}

// Read and discard results until no more.  Throw an exception if number of result sets read doesn't
// match number expected.  The result set count will be ignored if the expected count value
// passed in is negative.  A typical use of this routine is to verify that there are no more
// result sets to be read.
void CTPCC_DBLIB::DiscardNextResults(int iExpectedCount)
{
		int			iResultsRead = 0;
		RETCODE  rc;

		while (TRUE)
		{
			rc = dbresults(m_dbproc);
			if (rc == NO_MORE_RESULTS)
				break;
			if (rc == FAIL)
			{
				if (iExpectedCount >= 0)
					ThrowError(CDBLIBERR::eDbResults);
				else
					break;
			}

			DiscardNextRows(-1);
			iResultsRead++;
		}

		if ((iExpectedCount >= 0) &&
			(iExpectedCount != iResultsRead))
			ThrowError(CDBLIBERR::eWrongRowCount);
}

void CTPCC_DBLIB::StockLevel()
{
		int			iTryCount = 0;
		const BYTE *pData;

		ResetError();

		while (TRUE)
		{
			try
			{
				dbrpcinit(m_dbproc, "tpcc_stocklevel", 0);

				dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.StockLevel.w_id);		// @w_id smallint
				dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.StockLevel.d_id);		// @d_id tinyint
				dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.StockLevel.threshold);  // @threshhold smallint

				if (dbrpcexec(m_dbproc) == FAIL)
					ThrowError(CDBLIBERR::eDbRpcExec);

				if (dbresults(m_dbproc) != SUCCEED)
```

```
                                        ThrowError(CDBLIBERR::eDbResults);

                                if (dbnextrow(m_dbproc) != REG_ROW)
                                        ThrowError(CDBLIBERR::eDbNextRow);

                                if (pData=dbdata(m_dbproc, 1))
                                        m_txn.StockLevel.low_stock = *((long *) pData);

                                DiscardNextRows(0);
                                DiscardNextResults(0);

                                m_txn.StockLevel.exec_status_code = eOK;
                                return;
                        }
                        catch (CSQLERR *e)
                        {
                                if ((e->m_msgno != 1205) || (++iTryCount > iMaxRetries))
                                        throw;

                                // hit deadlock; backoff for increasingly longer period
                                delete e;
                                Sleep(10 * iTryCount);
                        }
                }               // while (TRUE)
}


void CTPCC_DBLIB::NewOrder()
{
        int                     i;
        DBINT           commit_flag;
        DBDATETIME      datetime;
        DBDATEREC       daterec;

        int                     iTryCount = 0;
        const BYTE *pData;

        ResetError();

        while (TRUE)
        {
                try
                {
                        dbrpcinit(m_dbproc, "tpcc_neworder", 0);

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.NewOrder.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.c_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_ol_cnt);

                        // check whether any order lines are for a remote warehouse
                        m_txn.NewOrder.o_all_local = 1;
                        for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
                        {
                                if (m_txn.NewOrder.OL[i].ol_supply_w_id != m_txn.NewOrder.w_id)
                                {
                                        m_txn.NewOrder.o_all_local = 0;  // at least one remote
warehouse
                                        break;
                                }
                        }
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_all_local);

                        for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
                        {
                                dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.OL[i].ol_i_id);
                                dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.NewOrder.OL[i].ol_supply_w_id);
                                dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.NewOrder.OL[i].ol_quantity);
                        }

                        if (dbrpcexec(m_dbproc) == FAIL)
                                ThrowError(CDBLIBERR::eDbRpcExec);


                        // Get order line results
                        m_txn.NewOrder.total_amount = 0;
                        for (i = 0; i<m_txn.NewOrder.o_ol_cnt; i++)
                        {
                                if (dbresults(m_dbproc) != SUCCEED)
                                        ThrowError(CDBLIBERR::eDbResults);

                                if (dbnumcols(m_dbproc) != 5)
                                        ThrowError(CDBLIBERR::eWrongNumCols);

                                if (dbnextrow(m_dbproc) != REG_ROW)
                                        ThrowError(CDBLIBERR::eDbNextRow);

                                if(pData=dbdata(m_dbproc, 1))
                                        UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name,pData,
dbdatlen(m_dbproc, 1));
                                if(pData=dbdata(m_dbproc, 2))
                                        m_txn.NewOrder.OL[i].ol_stock = (*(DBSMALLINT *)
pData);
                                if(pData=dbdata(m_dbproc, 3))
                                        UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic,
pData, dbdatlen(m_dbproc, 3));
                                if(pData=dbdata(m_dbproc, 4))
                                        dbconvert(m_dbproc, SQLNUMERIC, pData,
dbdatlen(m_dbproc,4),
                                        SQLFLT8, (BYTE *)&m_txn.NewOrder.OL[i].ol_i_price,
8);
                                if(pData=dbdata(m_dbproc, 5))
```

```
                                        dbconvert(m_dbproc, SQLNUMERIC, pData,
dbdatlen(m_dbproc,5),
                                        SQLFLT8, (BYTE *)&m_txn.NewOrder.OL[i].ol_amount,
8);

                                m_txn.NewOrder.total_amount = m_txn.NewOrder.total_amount +
m_txn.NewOrder.OL[i].ol_amount;

                                DiscardNextRows(0);
                        }


                        // get remaining values for w_tax, d_tax, o_id, c_last, c_discount, c_credit, o_entry_d,
commit_flag
                        if (dbresults(m_dbproc) != SUCCEED)
                                ThrowError(CDBLIBERR::eDbResults);

                        if (dbnextrow(m_dbproc) != REG_ROW)
                                ThrowError(CDBLIBERR::eDbNextRow);

                        if (dbnumcols(m_dbproc) != 8)
                                ThrowError(CDBLIBERR::eWrongNumCols);

                        if (pData=dbdata(m_dbproc, 1))

                                dbconvert(m_dbproc, SQLNUMERIC, pData, dbdatlen(m_dbproc,1),
SQLFLT8, (BYTE *)&m_txn.NewOrder.w_tax, 8);
                        if (pData=dbdata(m_dbproc, 2))

                                dbconvert(m_dbproc, SQLNUMERIC, pData, dbdatlen(m_dbproc,2),
SQLFLT8, (BYTE *)&m_txn.NewOrder.d_tax, 8);
                        if (pData=dbdata(m_dbproc, 3))
                                m_txn.NewOrder.o_id = (*(DBINT *) pData);
                        if (pData=dbdata(m_dbproc, 4))
                                UtilStrCpy(m_txn.NewOrder.c_last, pData, dbdatlen(m_dbproc, 4));
                        if (pData=dbdata(m_dbproc, 5))
                                dbconvert(m_dbproc, SQLNUMERIC, pData, dbdatlen(m_dbproc,5),
SQLFLT8, (BYTE *)&m_txn.NewOrder.c_discount, 8);
                        if (pData=dbdata(m_dbproc, 6))
                                UtilStrCpy(m_txn.NewOrder.c_credit, pData, dbdatlen(m_dbproc, 6));
                        if (pData=dbdata(m_dbproc, 7))
                        {
                                datetime = *((DBDATETIME *) pData);
                                dbdatecrack(m_dbproc, &daterec, &datetime);
                                m_txn.NewOrder.o_entry_d.year  = daterec.year;
                                m_txn.NewOrder.o_entry_d.month = daterec.month;
                                m_txn.NewOrder.o_entry_d.day   = daterec.day;
                                m_txn.NewOrder.o_entry_d.hour  = daterec.hour;
                                m_txn.NewOrder.o_entry_d.minute = daterec.minute;
                                m_txn.NewOrder.o_entry_d.second = daterec.second;
                        }
                        if (pData=dbdata(m_dbproc, 8))
                                commit_flag = (*(DBTINYINT *) pData);

                        DiscardNextRows(0);
                        DiscardNextResults(0);

                        if (commit_flag == 1)
                        {
                                m_txn.NewOrder.total_amount *= ((1 + m_txn.NewOrder.w_tax +
m_txn.NewOrder.d_tax) * (1 - m_txn.NewOrder.c_discount));
                                m_txn.NewOrder.exec_status_code = eOK;
                        }
                        else
                                m_txn.NewOrder.exec_status_code = eInvalidItem;

                        return;
                }
                catch (CSQLERR *e)
                {
                        if ((e->m_msgno != 1205) || (++iTryCount > iMaxRetries))
                                throw;

                        // hit deadlock; backoff for increasingly longer period
                        delete e;
                        Sleep(10 * iTryCount);
                }
        }       // while (TRUE)
}


void CTPCC_DBLIB::Payment()
{
        DBDATETIME      datetime;
        DBDATEREC       daterec;

        int                     iTryCount = 0;
        const BYTE *pData;

        ResetError();

        while (TRUE)
        {
                try
                {
                        dbrpcinit(m_dbproc, "tpcc_payment", 0);

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.Payment.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.Payment.c_w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLFLT8, -1, -1, (BYTE *)
&m_txn.Payment.h_amount);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.Payment.d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.Payment.c_d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.Payment.c_id);

                        // if customer id is zero, then payment is by name
```

```
                if (m_txn.Payment.c_id == 0)
                        dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.Payment.c_last),(unsigned char *)m_txn.Payment.c_last);

                if (dbrpcexec(m_dbproc) == FAIL)
                        ThrowError(CDBLIBERR::eDbRpcExec);

                if (dbresults(m_dbproc) != SUCCEED)
                        ThrowError(CDBLIBERR::eDbResults);

                if (dbnextrow(m_dbproc) != REG_ROW)
                        ThrowError(CDBLIBERR::eDbNextRow);

                if (dbnumcols(m_dbproc) != 27)
                        ThrowError(CDBLIBERR::eWrongNumCols);

                if (pData=dbdata(m_dbproc, 1))
                        m_txn.Payment.c_id = *((DBINT *) pData);
                if (pData=dbdata(m_dbproc, 2))
                        UtilStrCpy(m_txn.Payment.c_last,pData, dbdatlen(m_dbproc, 2));
                if (pData=dbdata(m_dbproc, 3))
                {
                        datetime = *((DBDATETIME *) pData);
                        dbdatecrack(m_dbproc, &daterec, &datetime);
                        m_txn.Payment.h_date.year  = daterec.year;
                        m_txn.Payment.h_date.month = daterec.month;
                        m_txn.Payment.h_date.day   = daterec.day;
                        m_txn.Payment.h_date.hour  = daterec.hour;
                        m_txn.Payment.h_date.minute= daterec.minute;
                        m_txn.Payment.h_date.second= daterec.second;
                }
                if (pData=dbdata(m_dbproc, 4))
                        UtilStrCpy(m_txn.Payment.w_street_1,pData, dbdatlen(m_dbproc, 4));
                if (pData=dbdata(m_dbproc, 5))
                        UtilStrCpy(m_txn.Payment.w_street_2,pData, dbdatlen(m_dbproc, 5));
                if (pData=dbdata(m_dbproc, 6))
                        UtilStrCpy(m_txn.Payment.w_city,pData, dbdatlen(m_dbproc, 6));
                if (pData=dbdata(m_dbproc, 7))
                        UtilStrCpy(m_txn.Payment.w_state,pData, dbdatlen(m_dbproc, 7));
                if (pData=dbdata(m_dbproc, 8))
                        UtilStrCpy(m_txn.Payment.w_zip,pData, dbdatlen(m_dbproc, 8));
                if (pData=dbdata(m_dbproc, 9))
                        UtilStrCpy(m_txn.Payment.d_street_1,pData, dbdatlen(m_dbproc, 9));
                if (pData=dbdata(m_dbproc, 10))
                        UtilStrCpy(m_txn.Payment.d_street_2,pData, dbdatlen(m_dbproc, 10));
                if (pData=dbdata(m_dbproc, 11))
                        UtilStrCpy(m_txn.Payment.d_city,pData, dbdatlen(m_dbproc, 11));
                if (pData=dbdata(m_dbproc, 12))
                        UtilStrCpy(m_txn.Payment.d_state,pData, dbdatlen(m_dbproc, 12));
                if (pData=dbdata(m_dbproc, 13))
                        UtilStrCpy(m_txn.Payment.d_zip,pData, dbdatlen(m_dbproc, 13));
                if (pData=dbdata(m_dbproc, 14))
                        UtilStrCpy(m_txn.Payment.c_first,pData, dbdatlen(m_dbproc, 14));
                if (pData=dbdata(m_dbproc, 15))
                        UtilStrCpy(m_txn.Payment.c_middle,pData, dbdatlen(m_dbproc, 15));
                if (pData=dbdata(m_dbproc, 16))
                        UtilStrCpy(m_txn.Payment.c_street_1,pData, dbdatlen(m_dbproc, 16));
                if (pData=dbdata(m_dbproc, 17))
                        UtilStrCpy(m_txn.Payment.c_street_2,pData, dbdatlen(m_dbproc, 17));
                if (pData=dbdata(m_dbproc, 18))
                        UtilStrCpy(m_txn.Payment.c_city,pData, dbdatlen(m_dbproc, 18));
                if (pData=dbdata(m_dbproc, 19))
                        UtilStrCpy(m_txn.Payment.c_state,pData, dbdatlen(m_dbproc, 19));
                if (pData=dbdata(m_dbproc, 20))
                        UtilStrCpy(m_txn.Payment.c_zip,pData, dbdatlen(m_dbproc, 20));
                if (pData=dbdata(m_dbproc, 21))
                        UtilStrCpy(m_txn.Payment.c_phone,pData, dbdatlen(m_dbproc, 21));
                if (pData=dbdata(m_dbproc, 22))
                {
                        datetime = *((DBDATETIME *) pData);
                        dbdatecrack(m_dbproc, &daterec, &datetime);
                        m_txn.Payment.c_since.year  = daterec.year;
                        m_txn.Payment.c_since.month = daterec.month;
                        m_txn.Payment.c_since.day   = daterec.day;
                        m_txn.Payment.c_since.hour  = daterec.hour;
                        m_txn.Payment.c_since.minute= daterec.minute;
                        m_txn.Payment.c_since.second= daterec.second;
                }
                if(pData=dbdata(m_dbproc, 23))
                        UtilStrCpy(m_txn.Payment.c_credit,pData, dbdatlen(m_dbproc, 23));
                if(pData=dbdata(m_dbproc, 24))
                        dbconvert(m_dbproc, SQLNUMERIC, pData, dbdatlen(m_dbproc,24),
SQLFLT8, (BYTE *)&m_txn.Payment.c_credit_lim,8);
                if(pData=dbdata(m_dbproc, 25))
                        dbconvert(m_dbproc, SQLNUMERIC, pData, dbdatlen(m_dbproc,25),
SQLFLT8, (BYTE *)&m_txn.Payment.c_discount,8);
                if(pData=dbdata(m_dbproc, 26))
                        dbconvert(m_dbproc, SQLNUMERIC, pData, dbdatlen(m_dbproc,26),
SQLFLT8, (BYTE *)&m_txn.Payment.c_balance,8);
                if(pData=dbdata(m_dbproc, 27))
                        UtilStrCpy(m_txn.Payment.c_data,pData, dbdatlen(m_dbproc, 27));

                DiscardNextRows(0);
                DiscardNextResults(0);


                if (m_txn.Payment.c_id == 0)
                        throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_INVALID_CUST);
                else
                        m_txn.Payment.exec_status_code = eOK;

                return;
        }
        catch (CSQLERR *e)
        {
                if ((e->m_msgno != 1205) || (++iTryCount > iMaxRetries))
                        throw;

                // hit deadlock; backoff for increasingly longer period
                delete e;
                Sleep(10 * iTryCount);
```

```
                }
        }       // while (TRUE)
}


void CTPCC_DBLIB::OrderStatus()
{
        int                             i;
        DBDATETIME      datetime;
        DBDATEREC       daterec;

        int                             iTryCount = 0;
        RETCODE         rc;
        const BYTE *pData;

        ResetError();

        while (TRUE)
        {
                try
                {
                        dbrpcinit(m_dbproc, "tpcc_orderstatus", 0);

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.OrderStatus.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.OrderStatus.d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.OrderStatus.c_id);

                        // if customer id is zero, then order status is by name
                        if (m_txn.OrderStatus.c_id == 0)
                                dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.OrderStatus.c_last), (unsigned char *)m_txn.OrderStatus.c_last);


                        if (dbrpcexec(m_dbproc) == FAIL)
                                ThrowError(CDBLIBERR::eDbRpcExec);

                        // Get order lines
                        if (dbresults(m_dbproc) != SUCCEED)
                        {
                                if ((m_DbLibErr == NULL) && (m_SqlErr == NULL))
                                        throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                                else
                                        ThrowError(CDBLIBERR::eDbResults);
                        }

                        if (dbnumcols(m_dbproc) != 5)
                                ThrowError(CDBLIBERR::eWrongNumCols);

                        i = 0;
                        while (TRUE)
                        {
                                rc = dbnextrow(m_dbproc);
                                if (rc == NO_MORE_ROWS)
                                        break;
                                if (rc != REG_ROW)
                                        ThrowError(CDBLIBERR::eDbNextRow);


                                if(pData=dbdata(m_dbproc, 1))
                                        m_txn.OrderStatus.OL[i].ol_supply_w_id=
(*(DBSMALLINT *) pData);
                                if(pData=dbdata(m_dbproc, 2))
                                        m_txn.OrderStatus.OL[i].ol_i_id= (*(DBINT *) pData);
                                if(pData=dbdata(m_dbproc, 3))
                                        m_txn.OrderStatus.OL[i].ol_quantity= (*(DBSMALLINT
*) pData);
                                if(pData=dbdata(m_dbproc, 4))
                                        dbconvert(m_dbproc, SQLNUMERIC, pData,
dbdatlen(m_dbproc,4),
                                                                SQLFLT8, (BYTE
*)&m_txn.OrderStatus.OL[i].ol_amount,8);
                                if(pData=dbdata(m_dbproc, 5))
                                {
                                        datetime = *((DBDATETIME *) pData);
                                        dbdatecrack(m_dbproc, &daterec, &datetime);
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.year  =
daterec.year;
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.month =
daterec.month;
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.day   =
daterec.day;
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.hour  =
daterec.hour;
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.minute=
daterec.minute;
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.second=
daterec.second;
                                }
                                i++;
                        }
                        m_txn.OrderStatus.o_ol_cnt = i;


                        if (dbresults(m_dbproc) != SUCCEED)
                                ThrowError(CDBLIBERR::eDbResults);

                        if (dbnextrow(m_dbproc) != REG_ROW)
                                ThrowError(CDBLIBERR::eDbNextRow);

                        if (dbnumcols(m_dbproc) != 8)
                                ThrowError(CDBLIBERR::eWrongNumCols);

                        if(pData=dbdata(m_dbproc, 1))
                                m_txn.OrderStatus.c_id = (*(DBINT *) pData);
                        if(pData=dbdata(m_dbproc, 2))
                                UtilStrCpy(m_txn.OrderStatus.c_last, pData, dbdatlen(m_dbproc,2));
                        if(pData=dbdata(m_dbproc, 3))
                                UtilStrCpy(m_txn.OrderStatus.c_first, pData, dbdatlen(m_dbproc,3));
```

```
                        if(pData=dbdata(m_dbproc, 4))
                                UtilStrCpy(m_txn.OrderStatus.c_middle,pData, dbdatlen(m_dbproc, 4));

                        if(pData=dbdata(m_dbproc, 5))
                        {
                                datetime = *((DBDATETIME *) pData);
                                dbdatecrack(m_dbproc, &daterec, &datetime);
                                m_txn.OrderStatus.o_entry_d.year  = daterec.year;
                                m_txn.OrderStatus.o_entry_d.month = daterec.month;
                                m_txn.OrderStatus.o_entry_d.day   = daterec.day;
                                m_txn.OrderStatus.o_entry_d.hour  = daterec.hour;
                                m_txn.OrderStatus.o_entry_d.minute= daterec.minute;
                                m_txn.OrderStatus.o_entry_d.second= daterec.second;
                        }
                        if(pData=dbdata(m_dbproc, 6))
                                m_txn.OrderStatus.o_carrier_id = (*(DBSMALLINT *) pData);
                        if(pData=dbdata(m_dbproc, 7))
                                dbconvert(m_dbproc, SQLNUMERIC, pData, dbdatlen(m_dbproc,7),
                                                SQLFLT8, (BYTE
*)&m_txn.OrderStatus.c_balance, 8);
                        if(pData=dbdata(m_dbproc, 8))
                                m_txn.OrderStatus.o_id = (*(DBINT *) pData);

                        DiscardNextRows(0);
                        DiscardNextResults(0);


                        if (m_txn.OrderStatus.o_ol_cnt == 0)
                                throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                        else if (m_txn.OrderStatus.c_id == 0 && m_txn.OrderStatus.c_last[0] == 0)
                                throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
                        else
                                m_txn.OrderStatus.exec_status_code = eOK;

                        return;
                }
                catch (CSQLERR *e)
                {
                        if ((e->m_msgno != 1205) || (++iTryCount > iMaxRetries))
                                throw;

                        // hit deadlock; backoff for increasingly longer period
                        delete e;
                        Sleep(10 * iTryCount);
                }
        }       // while (TRUE)
}


void CTPCC_DBLIB::Delivery()
{
        int                             i;
        int                             iTryCount = 0;
        const BYTE *pData;

        ResetError();

        while (TRUE)
        {
                try
                {
                        dbrpcinit(m_dbproc, "tpcc_delivery", 0);

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.Delivery.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.Delivery.o_carrier_id);

                        if (dbrpcexec(m_dbproc) == FAIL)
                                ThrowError(CDBLIBERR::eDbRpcExec);

                        if (dbresults(m_dbproc) != SUCCEED)
                                ThrowError(CDBLIBERR::eDbResults);

                        if (dbnextrow(m_dbproc) != REG_ROW)
                                ThrowError(CDBLIBERR::eDbNextRow);

                        if (dbnumcols(m_dbproc) != 10)
                                ThrowError(CDBLIBERR::eWrongNumCols);

                        for (i=0; i<10; i++)
                        {
                                if (pData = dbdata(m_dbproc, i+1))
                                        m_txn.Delivery.o_id[i]= *((DBINT *)pData);
                        }

                        DiscardNextRows(0);
                        DiscardNextResults(0);

                        m_txn.Delivery.exec_status_code= eOK;
                        return;
                }
                catch (CSQLERR *e)
                {
                        if ((e->m_msgno != 1205) || (++iTryCount > iMaxRetries))
                                throw;

                        // hit deadlock; backoff for increasingly longer period
                        delete e;
                        Sleep(10 * iTryCount);
                }
        }       // while (TRUE)
}

void CTPCC_DBLIB::ResetError()
{
        if (m_DbLibErr != NULL)
        {
                delete m_DbLibErr;
                m_DbLibErr = (CDBLIBERR*)NULL;
```

```
        }

        if (m_SqlErr != NULL)
        {
                delete m_SqlErr;
                m_SqlErr = (CSQLERR*)NULL;
        }
        return;
}
```

tpcc_dblib.h

```
/*      FILE:                   TPCC_DBLIB.H
 *                                      Microsoft TPC-C Kit Ver. 4.20.000
 *                                      Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                                      Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 *      PURPOSE:  Header file for TPC-C txn class implementation.
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */
#pragma once

#ifndef PDBPROCESS
#define DBPROCESS void   // dbprocess structure type
typedef DBPROCESS * PDBPROCESS;
#endif

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class CSQLERR : public CBaseErr
{
        public:

                CSQLERR(void)
                {
                        m_msgno = 0;
                        m_msgstate = 0;
                        m_severity = 0;
                        m_msgtext = NULL;
                };

                ~CSQLERR()
                {
                        delete [] m_msgtext;
                };

                int             m_msgno;
                int             m_msgstate;
                int             m_severity;
                char   *m_msgtext;

                int ErrorType() {return ERR_TYPE_SQL;};
                int ErrorNum() {return m_msgno;};
                char *ErrorText() {return m_msgtext;};

};

class CDBLIBERR : public CBaseErr
{
        public:
                enum ACTION
                {
                        eNone,
                        eUnknown,
                        eLogin,                         // error from dblogin
                        eDbOpen,                        // error from dbopen
                        eDbUse,                         // error from dbuse
                        eDbSqlExec,                     // error from dbsqlexec
                        eDbSet,                         // error from one of the dbset*
routines
                        eDbNextRow,                     // error from dbnextrow
                        eWrongRowCount,                 // more or less rows returned than expected
                        eWrongNumCols,                  // more or less columns returned than expected
                        eDbResults,                     // error from dbresults
                        eDbRpcExec,                     // error from dbrpcexec
                        eDbSetMaxProcs,                 // error from dbsetmaxprocs
                        eDbProcHandler                  // error from either dbprocerrhandle or
dbprocmsghandle
                };

                CDBLIBERR(ACTION eAction, int severity = 0, int dberror = 0, int oserr = 0)
                {
                        m_eAction = eAction;
                        m_severity = severity;
                        m_dberror = dberror;
                        m_oserr = oserr;

                        m_dberrstr = NULL;
                        m_oserrstr = NULL;
                };

                ~CDBLIBERR()
                {
                        delete [] m_dberrstr;
                        delete [] m_oserrstr;
                };

                ACTION  m_eAction;
                int             m_severity;
                int             m_dberror;
                int             m_oserr;
```

```
                char   *m_dberrstr;
                char   *m_oserrstr;

                int ErrorType() {return ERR_TYPE_DBLIB;};
                int ErrorNum() {return m_dberror;};
                char *ErrorText() {return m_dberrstr;};

};

class CTPCC_DBLIB_ERR : public CBaseErr
{
        public:
                enum CTPCC_DBLIB_ERRS
                {
                        ERR_WRONG_SP_VERSION = 1,    // "Wrong version of stored procs on database
server"
                        ERR_INVALID_CUST,                       // "Invalid Customer id,name."
                        ERR_NO_SUCH_ORDER                               // "No orders found
for customer."
                };

                CTPCC_DBLIB_ERR( int iErr ) { m_errno = iErr; };

                int                     m_errno;

                int ErrorType() {return ERR_TYPE_TPCC_DBLIB;};
                int ErrorNum() {return m_errno;};

                char *ErrorText();
};

class DllDecl CTPCC_DBLIB : public CTPCC_BASE
{
        private:
                // declare variables and private functions here...
                PDBPROCESS      m_dbproc;
                CDBLIBERR       *m_DbLibErr;                    // not allocated until needed
(maybe never)
                CSQLERR         *m_SqlErr;                      // not allocated until needed
(maybe never)
                int                             m_MaxRetries;                   // retry count on
deadlock

                void DiscardNextRows(int iExpectedCount);
                void DiscardNextResults(int iExpectedCount);
                void ThrowError( CDBLIBERR::ACTION eAction );
                void ResetError();

                union
                {
                        NEW_ORDER_DATA          NewOrder;
                        PAYMENT_DATA            Payment;
                        DELIVERY_DATA           Delivery;
                        STOCK_LEVEL_DATA StockLevel;
                        ORDER_STATUS_DATA       OrderStatus;
                } m_txn;

        public:
                CTPCC_DBLIB(LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost,
LPCSTR szDatabase );
                ~CTPCC_DBLIB(void);

                inline PNEW_ORDER_DATA          BuffAddr_NewOrder()             { return
&m_txn.NewOrder;   };
                inline PPAYMENT_DATA            BuffAddr_Payment()              { return
&m_txn.Payment;    };
                inline PDELIVERY_DATA           BuffAddr_Delivery()             { return
&m_txn.Delivery;   };
                inline PSTOCK_LEVEL_DATA        BuffAddr_StockLevel()   { return &m_txn.StockLevel; };
                inline PORDER_STATUS_DATA       BuffAddr_OrderStatus()  { return &m_txn.OrderStatus; };

                void NewOrder                   ();
                void Payment                    ();
                void Delivery                   ();
                void StockLevel                 ();
                void OrderStatus        ();


                // these are public because they must be called from the dblib err_handler and msg_hangler
                // outside of the class
                void SetDbLibError(int severity, int dberr, int oserr, LPCSTR dberrstr, LPCSTR oserrstr);
                void SetSqlError( int msgno, int msgstate, int severity, LPCSTR msgtext );

};

extern "C" DllDecl CTPCC_DBLIB* CTPCC_DBLIB_new
        ( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR szDatabase );

typedef CTPCC_DBLIB* (TYPE_CTPCC_DBLIB)(LPCSTR, LPCSTR, LPCSTR, LPCSTR, LPCSTR);
```

trans.h

```
/*              FILE:                   TRANS.H
 *                                              Microsoft TPC-C Kit Ver. 4.20.000
 *                                              Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                              Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 *              PURPOSE:  Header file for TPC-C structure templates.
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */
#pragma once

// String length constants
#define SERVER_NAME_LEN         20
#define DATABASE_NAME_LEN       20
```

```
#define USER_NAME_LEN            20
#define PASSWORD_LEN             20
#define TABLE_NAME_LEN           20
#define I_DATA_LEN               50
#define I_NAME_LEN               24
#define BRAND_LEN                1
#define LAST_NAME_LEN            16
#define W_NAME_LEN               10
#define ADDRESS_LEN              20
#define STATE_LEN                2
#define ZIP_LEN                  9
#define S_DIST_LEN               24
#define S_DATA_LEN               50
#define D_NAME_LEN               10
#define FIRST_NAME_LEN           16
#define MIDDLE_NAME_LEN          2
#define PHONE_LEN                16
#define DATETIME_LEN             30
#define CREDIT_LEN               2
#define C_DATA_LEN               250
#define H_DATA_LEN               24
#define DIST_INFO_LEN            24
#define MAX_OL_NEW_ORDER_ITEMS   15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN               25
#define OL_DIST_INFO_LEN                 24

// TIMESTAMP_STRUCT is provided by the ODBC header file sqltypes.h, but is not available
// when compiling with dblib, so redefined here.  Note: we are using the symbol "__SQLTYPES"
// (declared in sqltypes.h) as a way to determine if TIMESTAMP_STRUCT has been declared.
#ifndef __SQLTYPES
        typedef struct
        {
                short                   /* SQLSMALLINT */       year;
                unsigned short          /* SQLUSMALLINT */      month;
                unsigned short          /* SQLUSMALLINT */      day;
                unsigned short          /* SQLUSMALLINT */      hour;
                unsigned short          /* SQLUSMALLINT */      minute;
                unsigned short          /* SQLUSMALLINT */      second;
                unsigned long           /* SQLUINTEGER */       fraction;
        } TIMESTAMP_STRUCT;
#endif

// possible values for exec_status_code after transaction completes
enum EXEC_STATUS
{
        eOK,                    // 0    "Transaction commited."
        eInvalidItem,           // 1    "Item number is not valid."
        eDeliveryFailed         // 2    "Delivery Post Failed."
};

// transaction structures
typedef struct
{
        // input params
        short                           ol_supply_w_id;
        long                            ol_i_id;
        short                           ol_quantity;

        // output params
        char                            ol_i_name[I_NAME_LEN+1];
        char                            ol_brand_generic[BRAND_LEN+1];
        double                          ol_i_price;
        double                          ol_amount;
        short                           ol_stock;
} OL_NEW_ORDER_DATA;

typedef struct
{
        // input params
        short           w_id;
        short           d_id;
        long    c_id;
        short           o_ol_cnt;

        // output params
        EXEC_STATUS                     exec_status_code;
        char            c_last[LAST_NAME_LEN+1];
        char            c_credit[CREDIT_LEN+1];
        double          c_discount;
        double          w_tax;
        double          d_tax;
        long                    o_id;
        short                           o_commit_flag;
        TIMESTAMP_STRUCT o_entry_d;
        short           o_all_local;
        double          total_amount;
        OL_NEW_ORDER_DATA       OL[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA, *PNEW_ORDER_DATA;

typedef struct
{       // input params
        short                   w_id;
        short                   d_id;
        long                    c_id;
        short                   c_d_id;
        short                   c_w_id;
        double                  h_amount;
        char            c_last[LAST_NAME_LEN+1];

        // output params
        EXEC_STATUS                     exec_status_code;
        TIMESTAMP_STRUCT  h_date;
        char                    w_street_1[ADDRESS_LEN+1];
        char                    w_street_2[ADDRESS_LEN+1];
        char                    w_city[ADDRESS_LEN+1];
        char                    w_state[STATE_LEN+1];
        char                    w_zip[ZIP_LEN+1];
        char                    d_street_1[ADDRESS_LEN+1];
        char                    d_street_2[ADDRESS_LEN+1];
        char                    d_city[ADDRESS_LEN+1];
        char                    d_state[STATE_LEN+1];
```

```
        char                    d_zip[ZIP_LEN+1];
        char                    c_first[FIRST_NAME_LEN+1];
        char                    c_middle[MIDDLE_NAME_LEN+ 1];
        char                    c_street_1[ADDRESS_LEN+1];
        char                    c_street_2[ADDRESS_LEN+1];
        char                    c_city[ADDRESS_LEN+1];
        char                    c_state[STATE_LEN+1];
        char                    c_zip[ZIP_LEN+1];
        char                    c_phone[PHONE_LEN+1];
        TIMESTAMP_STRUCT  c_since;
        char                    c_credit[CREDIT_LEN+1];
        double                  c_credit_lim;
        double                  c_discount;
        double                  c_balance;
        char                    c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

typedef struct
{
        long                    ol_i_id;
        short                   ol_supply_w_id;
        short                   ol_quantity;
        double                  ol_amount;
        TIMESTAMP_STRUCT ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
{       // input params
        short           w_id;
        short           d_id;
        long            c_id;
        char            c_last[LAST_NAME_LEN+1];

        // output params
        EXEC_STATUS                             exec_status_code;
        char            c_first[FIRST_NAME_LEN+1];
        char            c_middle[MIDDLE_NAME_LEN+1];
        double           c_balance;
        long            o_id;
        TIMESTAMP_STRUCT  o_entry_d;
        short           o_carrier_id;
        OL_ORDER_STATUS_DATA OL[MAX_OL_ORDER_STATUS_ITEMS];
        short           o_ol_cnt;
} ORDER_STATUS_DATA, *PORDER_STATUS_DATA;


typedef struct
{       // input params
        short                   w_id;
        short                   o_carrier_id;

        // output params
        EXEC_STATUS                             exec_status_code;
        SYSTEMTIME                      queue_time;
        long                                    o_id[10];               // id's of delivered orders for
districts 1 to 10
} DELIVERY_DATA, *PDELIVERY_DATA;

//This structure is used for posting delivery transactions and for writing them to the delivery server.
typedef struct _DELIVERY_TRANSACTION
{
        SYSTEMTIME      queue;                  //time delivery transaction queued
        short           w_id;                   //delivery warehouse
        short           o_carrier_id;//carrier id
} DELIVERY_TRANSACTION;

typedef struct
{       // input params
        short                                   w_id;
        short                                   d_id;
        short                                   threshold;

        // output params
        EXEC_STATUS                             exec_status_code;
        long                                    low_stock;
} STOCK_LEVEL_DATA, *PSTOCK_LEVEL_DATA;
```

txnlog.h

```
/*      FILE:           TXNLOG.H
 *                              Microsoft TPC-C Kit Ver. 4.10.000
 *
 *                              NOTE: this file is RTE specific and should not be included
 *                              in Full Disclosure Reports.
 *
 *                              Copyright Microsoft, 1999
 *
 *      PURPOSE:  Structure definitions for logging deliverytxn completion stats.
 *      Contact:  Charles Levine (clevine@microsoft.com)
 */


  typedef struct _TXN_NEWORDER
        {
    BYTE  OL_Count;         //range 0 to 31
    BYTE  OL_Remote_Count;  //range 0 to 31
    WORD c_id;
    int                 o_id;
        } TXN_NEWORDER;

  typedef struct _TXN_PAYMENT
        {
    BYTE  CustByName;
    BYTE  IsRemote;
        } TXN_PAYMENT;

  typedef struct _TXN_ORDERSTATUS
        {
                BYTE    CustByName;
        } TXN_ORDERSTATUS;
```

```
typedef union _TXN_DETAILS
{
        TXN_NEWORDER        NewOrder;
        TXN_PAYMENT         Payment;
        TXN_ORDERSTATUS     OrderStatus;
} TXN_DETAILS;


// Common header for all records in txn log.  The TxnType field is
// a switch which identifies the particular variant.
#define TXN_REC_TYPE_CONTROL                    1       //
#define TXN_REC_TYPE_TPCC                       2       // replaces
TRANSACTION_TYPE_TPCC
#define TXN_REC_TYPE_TPCC_DELIV_DEF     3

typedef struct _TXN_RECORD_HEADER
{
        JULIAN_TIME     TxnStartT0;     // start of txn
        BYTE    TxnType;                // one of TXN_REC_TYPE_*
        BYTE    TxnSubType;                             // depends on TxnType
} TXN_RECORD_HEADER, *PTXN_RECORD_HEADER;


typedef struct _TXN_RECORD_CONTROL
{
        // common header; must exactly match TXN_RECORD_HEADER
        JULIAN_TIME     TxnStartT0;     // start of txn
        BYTE    TxnType;                // = TXN_REC_TYPE_CONTROL
        BYTE    TxnSubType;                             // depends on TxnType
        // end of common header

        DWORD   Len;                                    // number of bytes after this field
} TXN_RECORD_CONTROL, *PTXN_RECORD_CONTROL;


// TPC-C Txn Record Layout:
//
//'TxnStartT0' is a Julian timestamp corresponding to the moment the
//txn is sent to the SUT, i.e., beginning of response time.  Deltas
//are in milliseconds.  Note that if RTDelay > 0, then the txn was
//delayed by this amount.  The delay occurs at the beginning of the
//response time.  So if RTDelay > 0, then the txn was actually sent
//at TxnStartT0 + RTDelay.
//
//Graphically:
//
//  time -->
//
//    |--- Menu  ---|-- Keying  --|-- Response --|--- Think  --|
//     <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4  -> <- DeltaT3 ->
//                   ^
//                   ^  TxnStartT0
//
//RTDelay is the amount of response time delay included in DeltaT4.
//RTDelay is recorded per txn because this value can be changed on
//the fly, and so may vary from txn to txn.
//
//TxnStatus is the txn completion code.  It is used to indicate errors.
//For example, in the New Order txn, 1% of txns abort.  TxnStatus will
//reflect this.

        typedef struct _TXN_RECORD_TPCC
        {
                // common header; must exactly match TXN_RECORD_HEADER
                JULIAN_TIME     TxnStartT0;     // start of txn
                BYTE    TxnType;                // = TXN_REC_TYPE_TPCC
                BYTE    TxnSubType;                             // depends on TxnType
                // end of common header

                int     DeltaT1;                // menu time (ms)
                int     DeltaT2;                // keying time (ms)
                int     DeltaT3;                // think time (ms)
                int     DeltaT4;                // response time (ms)
                int     RTDelay;                // response time delay (ms)
                int             TxnError;               // error code providing more detail for
TxnStatus
                WORD    w_id;                           // warehouse ID
                BYTE    d_id;                           // assigned district ID for this thread
                BYTE    d_id_ThisTxn;           // district ID chosen for this particular
                BYTE    TxnStatus;              // completion status for txn to indicate errors
                BYTE    reserved;               // for word alignment
                TXN_DETAILS     TxnDetails;  //
        } TXN_RECORD_TPCC, *PTXN_RECORD_TPCC;

        // TPC-C Deferred Delivery Txn Record Layout:
        //
//Incorporating delivery transaction information into the above
        //structure would increase the size of TXN_DETAILS from 8 to 42 bytes.
//Hence, we store delivery transaction details in a separate structure.
//
        typedef struct _TXN_RECORD_TPCC_DELIV_DEF
        {
                // common header; must exactly match TXN_RECORD_HEADER
                JULIAN_TIME     TxnStartT0;     // start of txn
                BYTE    TxnType;                // = TXN_REC_TYPE_TPCC_DELIV_DEF
                BYTE    TxnSubType;                             // = 0
                // end of common header

                int     DeltaT4;                // response time (ms)
                int             DeltaTxnExec;           // execution time (ms)
                WORD    w_id;                           // warehouse ID
                BYTE    TxnStatus;              // completion status for txn to indicate errors
                BYTE    reserved;               // for word alignment
                short   o_carrier_id;   // carrier id
                long    o_id[10];               // returned delivery transaction ids
        } TXN_RECORD_TPCC_DELIV_DEF, *PTXN_RECORD_TPCC_DELIV_DEF;



#define         TXN_LOG_VERSION         1
#define         TXN_DATA_START          4096    // offset in log file where log records start
```

```
#define                 TXN_LOG_EYE_CATCHER     "BC"    // signature bytes at the start of log file


        ////////////////////////////////////////////////////////////////////////
        // The transaction log has a header as the first 4K block.
        //
        typedef struct _TXN_LOG_HEADER
        {
                char                    EyeCatcher[2];          // signature bytes; should always
be "BC"
                int                             LogVersion;             // set to
TXN_LOG_VERSION
                JULIAN_TIME             BeginTxnTS;                     // timestamp of first
(lowest) txn start
                JULIAN_TIME             EndTxnTS;               // timestamp of last (highest) txn
completion time
                int                             iRecCount;              // number of records
in log file
                BOOL                    bLogSorted;
                int                             iFileSize;              // file size in bytes

                // the record map provides a fast way to get close to a particular timestamp in a sorted log file.
//              struct
//              {
//                      JULIAN_TIME             TS;                             //
timestamp of record
//                      int                             iPos;                   // byte
position in file
//              }                               RecMap[RecMapSize];
//#define       RecMapSize              200

        } TXN_LOG_HEADER, *PTXN_LOG_HEADER;



#define READ_BUFFER_SIZE        64*1024
#define WRITE_BUFFER_SIZE        8*1024

#define NUM_READ_BUFFERS        1
#define NUM_WRITE_BUFFERS               2
#define MAX_NUM_BUFFERS                 2

// flags passed in to the constructor
#define TXN_LOG_WRITE           0x01
#define TXN_LOG_READ            0x02
#define TXN_LOG_SORTED          0x04

#define TXN_LOG_OS_ERROR 1
#define TXN_LOG_NOT_SORTED      2

#define SKIP_CTRL_RECS          1

class CTxnLog
{
        private:
                DWORD           iBufferSize;                    //buffer allocated size
                DWORD           iBytesFreeInBuffer;             //total bytes available for use in
buffer
                int                             iNumBuffers;                    //buffers
in use
                int                             iActiveBuffer;
//indicates which buffer is active: 0 or 1
                int                             iIoBuffer;                      //buffer
for any pending IO operation
                int                             iFilePointer;           //position in file.
                int                             iNextRec;                       //when
reading, ordinal value of next record

                // A "save point" is remembered each time GetNextRecord is called with a start time specified.
                // The next time it is called, if start time is after the save point, we start scanning from the
                // save point.  This is particularly useful in FindBestInterval, where the log is scanned repeatedly.
                JULIAN_TIME             SavePtTime;
                int                             iSavePtFilePointer;
                int                             iSavePtNextRec;

                JULIAN_TIME     lastTS;                                 //when
writing sorted output, used to verify records are sorted
                BOOL            bWrite;                                 //writing
log file

                BOOL                    bLogSorted;             // is log file sorted?
applies to both input and output
                JULIAN_TIME             BeginTxnTS;                     //
timestamp of first (lowest) txn start
                JULIAN_TIME             EndTxnTS;               // timestamp of last
(highest) txn completion time
                int                             iRecCount;              //
number of records in log file

                BYTE            *pCurrent;                              //ptr to current buffer
                BYTE            *pBuffer[MAX_NUM_BUFFERS];

                PTXN_RECORD_HEADER      *TxnArray;      //transaction record pointer array
for sort

                DWORD           dwError;
                HANDLE          hTxnFile;                               //handle to log file
                HANDLE          hMapFile;                               //map file used when
sorting the log
                HANDLE          hIoComplete;                            //event to signify that
there are no pending IOs
                HANDLE          hLogFileIo;                             //event to signal the
IO thread to write the inactive buffer

                Spinlock        Spin;                                   //spin lock to protect
the txn log file buffers

                int Write(BYTE *ptr, DWORD Size);
                static void LogFileIO(CTxnLog*);

        public:
```

```
                        CTxnLog::CTxnLog(LPCTSTRszFileName, DWORD dwOpts);
                        ~CTxnLog(void);

                        int WriteToLog(PTXN_RECORD_TPCC pTxnRcrd);
                        int WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF pTxnRcrd);
                        int WriteToLog(PTXN_RECORD_CONTROL pCtrlRec);
                        int WriteToLog(PTXN_RECORD_HEADER pCtrlRec);

                        int WriteCtrlRecToLog(BYTE SubType, LPTSTR lpStr, DWORD dwLen);

                        void CloseTransactionLogFile(void);

                        PTXN_RECORD_HEADER GetNextRecord(BOOL bSkipCtrlRecs = FALSE);
                        PTXN_RECORD_HEADER GetNextRecord(JULIAN_TIME SeekTimeT0, BOOL
bSkipCtrlRecs = FALSE);

                        int Sort(void);
                        PTXN_RECORD_HEADER GetSortedRecord(int index);

                        inline BOOL IsSorted(void) { return bLogSorted; };
                        inline JULIAN_TIME BeginTS(void) { return BeginTxnTS; };
                        inline JULIAN_TIME EndTS(void) { return EndTxnTS; };
                        inline int RecordCount(void) { return iRecCount; };
};

class CTXNLOG_ERR : public CBaseErr
{
        public:
                        enum CTPCC_DBLIB_ERRS
                        {
                                ERR_BAD_FILE_FORMAT = 1,        // "File format is invalid."
                                ERR_UNKNOWN_LOG_VERSION, // "Log file version is unknown."
                                ERR_BROKEN_LOG_FILE,                    // "Log file is broken."
                                ERR_LOG_NOT_SORTED,                             // "Log file is not
sorted"
                                ERR_INVALID_TIME_SEQ,                   // "Internal Error: Record Time
Sequence invalid."
                        };

                        CTXNLOG_ERR( int iErr ) { m_errno = iErr; };

                        int                     m_errno;

                        int ErrorType() {return ERR_TYPE_TXNLOG;};
                        int ErrorNum() {return m_errno;};

                        // TODO: need to complete...
                        char *ErrorText() {return "";};
};
```

txn_base.h

```
/*      FILE:                   TXN_BASE.H
 *                                      Microsoft TPC-C Kit Ver. 4.20.000
 *                                                      Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                                                      Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 *       PURPOSE:  Header file for TPC-C txn class implementation.
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */

#pragma once

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class DllDecl CTPCC_BASE
{
        public:
                        CTPCC_BASE(void) {};
                        virtual ~CTPCC_BASE(void) {};

                        virtual PNEW_ORDER_DATA                 BuffAddr_NewOrder()             = 0;
                        virtual PPAYMENT_DATA                   BuffAddr_Payment()              = 0;
                        virtual PDELIVERY_DATA                  BuffAddr_Delivery()             = 0;
                        virtual PSTOCK_LEVEL_DATA       BuffAddr_StockLevel()   = 0;
                        virtual PORDER_STATUS_DATA      BuffAddr_OrderStatus()  = 0;

                        virtual void NewOrder           () = 0;
                        virtual void Payment            () = 0;
                        virtual void Delivery           () = 0;
                        virtual void StockLevel         () = 0;
                        virtual void OrderStatus   () = 0;
};
```

webclnt.dsp

```
# Microsoft Developer Studio Project File - Name="webclnt" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=webclnt - Win32 Release
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
```

```
!MESSAGE by defining the macroCFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak" CFG="webclnt - Win32 Release"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "webclnt - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "webclnt - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "webclnt - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir ".\Release"
# PROP BASE Intermediate_Dir ".\Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\Release"
# PROP Intermediate_Dir ".\Release"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386

!ELSEIF  "$(CFG)" == "webclnt - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\Debug"
# PROP Intermediate_Dir ".\Debug"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /debug /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /debug /machine:I386

!ENDIF

# Begin Target

# Name "webclnt - Win32 Release"
# Name "webclnt - Win32 Debug"
# End Target
# End Project
```

webclnt.dsw

```
Microsoft Developer Studio Workspace File, Format Version 5.00
# WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

###############################################################################

Project: "db_dblib_dll"=.\db_dblib_dll\db_dblib_dll.dsp- Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "db_odbc_dll"=.\db_odbc_dll\db_odbc_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################
```

```
Project: "install"=.\install\install.dsp- Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name isapi_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tuxapp
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_com_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_tuxedo_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_all
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
}}}

###############################################################################

Project: "isapi_dll"=.\isapi_dll\isapi_dll.dsp- Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_tuxedo_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_encina_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_com_dll
    End Project Dependency
}}}

###############################################################################

Project: "tm_com_dll"=.\tm_com_dll\tm_com_dll.dsp- Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_all
    End Project Dependency
}}}

###############################################################################

Project: "tm_encina_dll"=.\tm_encina_dll\tm_encina_dll.dsp- Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "tm_tuxedo_dll"=.\tm_tuxedo_dll\tm_tuxedo_dll.dsp- Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "tpcc_com_all"=.\tpcc_com_all\tpcc_com_all.dsp- Package Owner=<4>

Package=<5>
{{{
}}}
```

```
Package=<4>
{{{
   Begin Project Dependency
   Project_Dep_Name tpcc_com_ps
   End Project Dependency
}}}

###############################################################################

Project: "tpcc_com_ps"=.\tpcc_com_ps\tpcc_com_ps.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "tuxapp"=.\tuxapp\tuxapp.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
   Begin Project Dependency
   Project_Dep_Name db_dblib_dll
   End Project Dependency
   Begin Project Dependency
   Project_Dep_Name db_odbc_dll
   End Project Dependency
}}}

###############################################################################

Global:

Package=<5>
{{{
}}}

Package=<3>
{{{
}}}

###############################################################################
```

Appendix B: Database Design

backup.sql

```
-- File:    BACKUP.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates backup of tpcc database


declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

dump database tpcc to tpccback1, tpccback2, tpccback3 with init, stats = 1

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

backupdev.sql

```
-- File:    BACKUPDEVB.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates tpcc database Backup Devices


use master
go

-- create backup devices

exec sp_addumpdevice 'disk','tpccback1','Z:\tpccback1.dmp'
exec sp_addumpdevice 'disk','tpccback2','Z:\tpccback2.dmp'
exec sp_addumpdevice 'disk','tpccback3','Z:\tpccback3.dmp'

go
```

createdb.sql

```
-- File:    CREATEDB.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates tpcc database and backup files

use master
go

--          Create temporary table for timing

if exists ( select name from sysobjects where name = 'tpcc_timer' )
        drop table tpcc_timer
go

create table tpcc_timer
(
        start_date                                      char(30),
        end_date                                        char(30)
)

insert  into tpcc_timer values (0,0)
go

--          Store starting time

update  tpcc_timer
set     start_date    = (select convert(char(30), getdate(),9))
go

--  create main database files

CREATE DATABASE tpcc
ON PRIMARY
(       NAME                    = MSSQL70_tpcc_root,
        FILENAME= "C:\MSSQL70_tpcc_root.mdf",
        SIZE                    = 8MB,
        FILEGROWTH              =0),
FILEGROUP       MSSQL70_misc_fg
(       NAME                    = MSSQL70_misc1,
        FILENAME= "F:",
        SIZE                    = 9600MB,
        FILEGROWTH              = 0),
(       NAME                    = MSSQL70_misc2,
        FILENAME= "H:",
        SIZE                    = 9600MB,
        FILEGROWTH              = 0),
(       NAME                    = MSSQL70_misc3,
        FILENAME= "J:",
        SIZE                    = 9600MB,
        FILEGROWTH              = 0),
(       NAME                    = MSSQL70_misc4,
        FILENAME= "L:",
        SIZE                    = 9600MB,
        FILEGROWTH              = 0),
(       NAME                    = MSSQL70_misc5,
        FILENAME= "N:",
        SIZE                    = 9600MB,
        FILEGROWTH              = 0),
(       NAME                    = MSSQL70_misc6,
        FILENAME= "P:",
        SIZE                    = 9600MB,
```

```
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc7,
        FILENAME = "R:",
        SIZE            = 9600MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc8,
        FILENAME = "T:",
        SIZE            = 9600MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc9,
        FILENAME = "V:",
        SIZE            = 9600MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_misc10,
        FILENAME = "X:",
        SIZE            = 9600MB,
        FILEGROWTH      = 0),
FILEGROUP       MSSQL70_cs_fg
(       NAME            = MSSQL70_cs1,
        FILENAME = "E:",
        SIZE            = 19200MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs2,
        FILENAME = "G:",
        SIZE            = 19200MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs3,
        FILENAME = "I:",
        SIZE            = 19200MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs4,
        FILENAME = "K:",
        SIZE            = 19200MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs5,
        FILENAME = "M:",
        SIZE            = 19200MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs6,
        FILENAME = "O:",
        SIZE            = 19200MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs7,
        FILENAME = "Q:",
        SIZE            = 19200MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs8,
        FILENAME = "S:",
        SIZE            = 19200MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs9,
        FILENAME = "U:",
        SIZE            = 19200MB,
        FILEGROWTH      = 0),
(       NAME            = MSSQL70_cs10,
        FILENAME = "W:",
        SIZE            = 19200MB,
        FILEGROWTH      = 0)
LOG ON
(       NAME            =MSSQL70_tpcc_log,
        FILENAME ="Y:",
        SIZE            =60000MB,
        FILEGROWTH      =0)
go

-- Store ending time
update  tpcc_timer
set     end_date    = (select convert(char(30), getdate(),9))
go

select "Elapsed time (in seconds): ", datediff(second,(select start_date from tpcc_timer),(select end_date from tpcc_timer))

--      remove temporary table

if exists ( select name from sysobjects where name = 'tpcc_timer' )
        drop table tpcc_timer
go
```

tables.sql

```
--  File:    TABLES.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.20
--           Copyright Microsoft, 1999
--  Purpose:  Creates TPC-C tables


use tpcc
go

--
--  Remove all existing TPC-C tables
--

if exists ( select name from sysobjects where name = 'warehouse' )
        drop table warehouse
go
if exists ( select name from sysobjects where name = 'district' )
        drop table district
go
if exists ( select name from sysobjects where name = 'customer' )
        drop table customer
go
if exists ( select name from sysobjects where name = 'history' )
        drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
        drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
        drop table orders
```

```
go
if exists ( select name from sysobjects where name = 'order_line')
        drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
        drop table item
go
if exists ( select name from sysobjects where name = 'stock' )
        drop table stock
go

--
--  Create new tables
--

create table warehouse
(
        w_id                            smallint,
        w_name                          char(10),
        w_street_1              char(20),
        w_street_2              char(20),
        w_city                          char(20),
        w_state                         char(2),
        w_zip                           char(9),
        w_tax                           numeric(4,4),
        w_ytd                           numeric(12,2)
) on MSSQL70_misc_fg
go

create table district
(
        d_id                            tinyint,
        d_w_id                          smallint,
        d_name                          char(10),
        d_street_1              char(20),
        d_street_2              char(20),
        d_city                          char(20),
        d_state                         char(2),
        d_zip                           char(9),
        d_tax                           numeric(4,4),
        d_ytd                           numeric(12,2),
        d_next_o_id             int
) on MSSQL70_misc_fg
go

create table customer
(
        c_id                            int,
        c_d_id                          tinyint,
        c_w_id                          smallint,
        c_first                         char(16),
        c_middle                char(2),
        c_last                          char(16),
        c_street_1              char(20),
        c_street_2              char(20),
        c_city                          char(20),
        c_state                         char(2),
        c_zip                           char(9),
        c_phone                         char(16),
        c_since                         datetime,
        c_credit                char(2),
        c_credit_lim    numeric(12,2),
        c_discount              numeric(4,4),
        c_balance               numeric(12,2),
        c_ytd_payment           numeric(12,2),
        c_payment_cnt           smallint,
        c_delivery_cnt          smallint,
        c_data                          char(500)
) on MSSQL70_cs_fg
go

create table history
(
        h_c_id                          int,
        h_c_d_id                tinyint,
        h_c_w_id                smallint,
        h_d_id                          tinyint,
        h_w_id                          smallint,
        h_date                          datetime,
        h_amount                numeric(6,2),
        h_data                          char(24)
) on MSSQL70_misc_fg
go

create table new_order
(
        no_o_id                         int,
        no_d_id                         tinyint,
        no_w_id                         smallint
) on MSSQL70_misc_fg
go

create table orders
(
        o_id                            int,
        o_d_id                          tinyint,
        o_w_id                          smallint,
        o_c_id                          int,
        o_entry_d               datetime,
        o_carrier_id    tinyint,
        o_ol_cnt                tinyint,
        o_all_local             tinyint
) on MSSQL70_misc_fg
go

create table order_line
(
        ol_o_id                         int,
        ol_d_id                         tinyint,
        ol_w_id                         smallint,
        ol_number               tinyint,
        ol_i_id                         int,
```

```
        ol_supply_w_id          smallint,
        ol_delivery_d           datetime,
        ol_quantity             smallint,
        ol_amount               numeric(6,2),
        ol_dist_info    char(24)
) on MSSQL70_misc_fg
go

create table item
(
        i_id                            int,
        i_im_id                         int,
        i_name                          char(24),
        i_price                         numeric(5,2),
        i_data                          char(50)
) on MSSQL70_misc_fg
go

create table stock
(
        s_i_id                          int,
        s_w_id                          smallint,
        s_quantity              smallint,
        s_dist_01               char(24),
        s_dist_02               char(24),
        s_dist_03               char(24),
        s_dist_04               char(24),
        s_dist_05               char(24),
        s_dist_06               char(24),
        s_dist_07               char(24),
        s_dist_08               char(24),
        s_dist_09               char(24),
        s_dist_10               char(24),
        s_ytd                           int,
        s_order_cnt             smallint,
        s_remote_cnt            smallint,
        s_data                          char(50)
) on MSSQL70_cs_fg
go
```

dbopt1.sql

```
-- File:    DBOPT1.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Sets database options for data load


use master
go

exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
go

use tpcc
go

checkpoint
go
```

dbopt2.sql

```
-- File:    DBOPT2.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Resets database options after data load


use master
go

sp_dboption tpcc,'select ',false
go

sp_dboption tpcc,'trunc. ',false
go

use tpcc
go

checkpoint
go

sp_configure allow,1
go

reconfigure with override
go

/*                                      */
/* Set option values for user-defined indexes */
/*                                      */

sp_indexoption          'customer',   'AllowPageLocks',       FALSE
go
sp_indexoption          'district',   'AllowPageLocks',       FALSE
go
sp_indexoption          'warehouse', 'AllowPageLocks',        FALSE
go
sp_indexoption          'stock',      'AllowPageLocks',       FALSE
go
sp_indexoption          'order_line', 'AllowRowLocks',        FALSE
go
sp_indexoption          'orders',     'AllowRowLocks',        FALSE
go
sp_indexoption          'new_order', 'AllowRowLocks',         FALSE
go
```

```
sp_indexoption          'item',                 'AllowRowLocks',        FALSE
go
sp_indexoption          'item',                 'AllowPageLocks',       FALSE
go

Print ' '
Print '******************'
Print 'Pre-specified Locking Hierarchy:'
Print '   Lockflag = 0 ==> No pre-specified hierarchy'
Print '   Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '   Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '   Lockflag = 3 ==> Lock at Table-level'
Print ' '

select      name,lockflags
from        sysindexes
where       object_id("warehouse")   = id or
            object_id("district")    = id or
            object_id("customer")    = id or
            object_id("stock")       = id or
            object_id("orders")      = id or
            object_id("order_line")  = id or
            object_id("history")     = id or
            object_id("new_order")   = id or
            object_id("item")        = id
order       by lockflags asc
go

sp_configure allow,0
go

reconfigure with override
go

exec sp_dboption tpcc,   'auto update statistics',   FALSE
exec sp_dboption tpcc,   'auto create statistics',   FALSE
go

exec sp_tableoption      "district",    "pintable",true
exec sp_tableoption      "warehouse",               "pintable",true
exec sp_tableoption      "new_order",               "pintable",true
exec sp_tableoption      "item",                    "pintable",true
go
```

idxcusnc.sql

```
-- File:    IDXCUSNC.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates non-clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'customer_nc1' )
            drop index customer.customer_nc1

create unique nonclustered index customer_nc1 on customer(c_w_id, c_d_id, c_last, c_first, c_id)
            on MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxcuscl.sql

```
-- File:    IDXCUSCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'customer_c1' )
            drop index customer.customer_c1

create unique clustered index customer_c1 on customer(c_w_id, c_d_id, c_id)
            on MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxdiscl.sql

```
-- File:    IDXDISCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates clustered index on district table
```

```
use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'district_c1' )
        drop index district.district_c1

create unique clustered index  district_c1 on district(d_w_id, d_id)
        with fillfactor=100 on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxitmcl.sql

```
-- File:    IDXITMCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates clustered index on item table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'item_c1')
        drop index item.item_c1

create unique clustered index item_c1 on item(i_id)
        on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxnodcl.sql

```
-- File:    IDXNODCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates clustered index on new_order table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'new_order_c1' )
        drop index new_order.new_order_c1

create unique clustered index new_order_c1 on new_order(no_w_id, no_d_id, no_o_id)
        on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxodlcl.sql

```
-- File:    IDXODLCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates clustered index on order_line table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'order_line_c1')
        drop index order_line.order_line_c1

create unique clustered index order_line_c1 on order_line(ol_w_id, ol_d_id, ol_o_id, ol_number)
        on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordnc.sql

```
-- File:    IDXORDNC.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates non-clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'orders_nc1' )
                drop index orders.orders_nc1

create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
                on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordcl.sql

```
-- File:    IDXORDCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'orders_c1' )
                drop index orders.orders_c1

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
                on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxstkcl.sql

```
-- File:    IDXSTKCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates clustered index on stock table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'stock_c1' )
                drop index stock.stock_c1

create unique clustered index stock_c1 on stock(s_i_id, s_w_id)
                on MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxwarcl.sql

```
-- File:    IDXWARCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20
--          Copyright Microsoft, 1999
-- Purpose:  Creates clustered index on warehouse table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'warehouse_c1' )
                drop index warehouse.warehouse_c1

create unique clustered index warehouse_c1 on warehouse(w_id)
go
```

```
        with fillfactor=100 on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

Stored Procedures

neword.sql

```
-- File:    NEWORD.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--          Copyright Microsoft, 1999
-- Purpose: Creates new order transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_neworder" )
        drop procedure tpcc_neworder
go

create proc tpcc_neworder
                                        @w_id           smallint,
                                        @d_id           tinyint,
                                        @c_id           int,
                                        @o_ol_cnt       tinyint,
                                        @o_all_local    tinyint,
                                        @i_id1  int = 0, @s_w_id1  smallint = 0, @ol_qty1
smallint = 0,
                                        @i_id2  int = 0, @s_w_id2  smallint = 0, @ol_qty2
smallint = 0,
                                        @i_id3  int = 0, @s_w_id3  smallint = 0, @ol_qty3
smallint = 0,
                                        @i_id4  int = 0, @s_w_id4  smallint = 0, @ol_qty4
smallint = 0,
                                        @i_id5  int = 0, @s_w_id5  smallint = 0, @ol_qty5
smallint = 0,
                                        @i_id6  int = 0, @s_w_id6  smallint = 0, @ol_qty6
smallint = 0,
                                        @i_id7  int = 0, @s_w_id7  smallint = 0, @ol_qty7
smallint = 0,
                                        @i_id8  int = 0, @s_w_id8  smallint = 0, @ol_qty8
smallint = 0,
                                        @i_id9  int = 0, @s_w_id9  smallint = 0, @ol_qty9
smallint = 0,
                                        @i_id10 int = 0, @s_w_id10 smallint = 0, @ol_qty10
smallint = 0,
                                        @i_id11 int = 0, @s_w_id11 smallint = 0, @ol_qty11
smallint = 0,
                                        @i_id12 int = 0, @s_w_id12 smallint = 0, @ol_qty12
smallint = 0,
                                        @i_id13 int = 0, @s_w_id13 smallint = 0, @ol_qty13
smallint = 0,
                                        @i_id14 int = 0, @s_w_id14 smallint = 0, @ol_qty14
smallint = 0,
                                        @i_id15 int = 0, @s_w_id15 smallint = 0, @ol_qty15
smallint = 0

as
declare         @w_tax          numeric(4,4),
        @d_tax          numeric(4,4),
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     numeric(4,4),
        @i_price        numeric(5,2),
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
                @li_no          int,
                @o_id                           int,
                @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      smallint,
        @li_qty         smallint,
                @ol_numberint,
                @c_id_localint

begin

begin transaction n

-- get district tax and next availible order id and update
-- plus initialize local variables

                update          district
                set             @d_tax       = d_tax,
                                @o_id        = d_next_o_id,
                                d_next_o_id  = d_next_o_id + 1,
                                @o_entry_d   = getdate(),
                                @li_no       = 0,
                                @commit_flag= 1
                where           d_w_id       = @w_id and
```

```
                        d_id        = @d_id

-- process orderlines

                while (@li_no < @o_ol_cnt)
                begin

                        select @li_no = @li_no + 1

-- set i_id, s_w_id, and qty for this lineitem

                        select      @li_id = case @li_no
                                        when 1 then @i_id1
                                        when 2 then @i_id2
                                        when 3 then @i_id3
                                        when 4 then @i_id4
                                        when 5 then @i_id5
                                        when 6 then @i_id6
                                        when 7 then @i_id7
                                        when 8 then @i_id8
                                        when 9 then @i_id9
                                        when 10 then @i_id10
                                        when 11 then @i_id11
                                        when 12 then @i_id12
                                        when 13 then @i_id13
                                        when 14 then @i_id14
                                        when 15 then @i_id15
                                        end,

                                @li_s_w_id = case @li_no
                                        when 1 then @s_w_id1
                                        when 2 then @s_w_id2
                                        when 3 then @s_w_id3
                                        when 4 then @s_w_id4
                                        when 5 then @s_w_id5
                                        when 6 then @s_w_id6
                                        when 7 then @s_w_id7
                                        when 8 then @s_w_id8
                                        when 9 then @s_w_id9
                                        when 10 then @s_w_id10
                                        when 11 then @s_w_id11
                                        when 12 then @s_w_id12
                                        when 13 then @s_w_id13
                                        when 14 then @s_w_id14
                                        when 15 then @s_w_id15
                                        end,

                                @li_qty = case  @li_no
                                        when 1 then @ol_qty1
                                        when 2 then @ol_qty2
                                        when 3 then @ol_qty3
                                        when 4 then @ol_qty4
                                        when 5 then @ol_qty5
                                        when 6 then @ol_qty6
                                        when 7 then @ol_qty7
                                        when 8 then @ol_qty8
                                        when 9 then @ol_qty9
                                        when 10 then @ol_qty10
                                        when 11 then @ol_qty11
                                        when 12 then @ol_qty12
                                        when 13 then @ol_qty13
                                        when 14 then @ol_qty14
                                        when 15 then @ol_qty15
                                        end

-- get item data (no one updates item)

                        select      @i_price = i_price,
                                    @i_name  = i_name,
                                    @i_data  = i_data
                        from        item (tablock repeatableread)
                        where       i_id = @li_id

-- update stock values

                        update      stock
                        set         s_ytd                   = s_ytd + @li_qty,
                                    @s_quantity= s_quantity   = s_quantity - @li_qty +
                                                                    case when (s_quantity - @li_qty < 10) then
91 else 0 end,
                                    s_order_cnt  = s_order_cnt + 1,
                                    s_remote_cnt            = s_remote_cnt + case when (@li_s_w_id = @w_id) then 0
else 1 end,
                                    @s_data                 = s_data,
                                    @s_dist                 = case @d_id
                                      when 1  then s_dist_01
                                      when 2  then s_dist_02
                                      when 3  then s_dist_03
                                      when 4  then s_dist_04
                                      when 5  then s_dist_05
                                      when 6  then s_dist_06
                                      when 7  then s_dist_07
                                      when 8  then s_dist_08
                                      when 9  then s_dist_09
                                      when 10 then s_dist_10
                                      end
                        where       s_i_id                  = @li_id and
                                    s_w_id                  = @li_s_w_id

-- if there actually is a stock (and item) with these ids, go to work

                        if (@@rowcount > 0)
                        begin

-- insert order_line data (using data from item and stock)

                                insert into order_line values(@o_id,
                                                                @d_id,
                                                                @w_id,
                                                                @li_no,
                                                                @li_id,
                                                                @li_s_w_id,
```

```
                                                        "dec 31, 1899",
                                                        @li_qty,
                                                        @i_price * @li_qty,
                                                        @s_dist)

-- send line-item data to client

                                        select          @i_name,
                                                        @s_quantity,
                                                        b_g = case when ( (patindex("%ORIGINAL%",@i_data)> 0) and
                                                                          (patindex("%ORIGINAL%",@s_data)> 0) )
                                                            then "B" else "G" end,
                                                        @i_price,
                                                        @i_price * @li_qty

                                end
                                else
                                begin

-- no item (or stock) found - triggers rollback condition

                                        select "",0,"",0,0
                                        select @commit_flag= 0

                                end
                        end

-- get customer last name, discount, and credit rating

                        select          @c_last     = c_last,
                                        @c_discount = c_discount,
                                        @c_credit   = c_credit,
                                        @c_id_local = c_id
                        from            customer (repeatableread)
                        where           c_id                    = @c_id and
                                        c_w_id                  = @w_id and
                                        c_d_id                  = @d_id

-- insert fresh row into orders table

                        insert into orders values ( @o_id,
                                                                @d_id,
                                                                @w_id,
                                                                @c_id_local,
                                                                @o_entry_d,
                                                                0,
                                                                @o_ol_cnt,
                                                                @o_all_local)

-- insert corresponding row into new-order table

                        insert into new_order values (          @o_id,
                                                                @d_id,
                                                                @w_id)

-- select warehouse tax

                        select          @w_tax      = w_tax
                        from            warehouse (repeatableread)
                        where           w_id            = @w_id

                        if (@commit_flag= 1)
                                commit transaction n
                        else

-- all that work for nuthin!!!

                                rollback transaction n

-- return order data to client

                        select          @w_tax,
                                        @d_tax,
                                        @o_id,
                                        @c_last,
                                        @c_discount,
                                        @c_credit,
                                        @o_entry_d,
                                        @commit_flag

end

go
```

payment.sql

```
-- File:    PAYMENT.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--          Copyright Microsoft, 1999
-- Purpose:  Creates payment transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_payment" )
        drop procedure tpcc_payment
go

create proc tpcc_payment@w_id           smallint,
                                                @c_w_id         smallint,
                                                @h_amount       numeric(6,2),
                                                @d_id           tinyint,
                                                @c_d_id         tinyint,
                                                @c_id           int,
                                                @c_last         char(16) = ""


as
```

```
declare  @w_street_1     char(20),
         @w_street_2     char(20),
         @w_city         char(20),
         @w_state        char(2),
         @w_zip          char(9),
         @w_name         char(10),
         @d_street_1     char(20),
         @d_street_2     char(20),
         @d_city         char(20),
         @d_state        char(2),
         @d_zip          char(9),
         @d_name         char(10),
         @c_first        char(16),
         @c_middle       char(2),
         @c_street_1     char(20),
         @c_street_2     char(20),
         @c_city         char(20),
         @c_state        char(2),
         @c_zip          char(9),
         @c_phone        char(16),
         @c_since        datetime,
         @c_credit       char(2),
         @c_credit_lim   numeric(12,2),
         @c_balance      numeric(12,2),
         @c_discount     numeric(4,4),
         @data           char(500),
         @c_data         char(500),
         @datetime       datetime,
         @w_ytd          numeric(12,2),
         @d_ytd          numeric(12,2),
         @cnt            smallint,
         @val            smallint,
         @screen_data          char(200),
                 @d_id_local             tinyint,
                 @w_id_local             smallint,
                 @c_id_local             int

select @screen_data = ""

begin tran p

--  get payment date

                 select          @datetime = getdate()

                 if (@c_id = 0)
                 begin

--  get customer id and info using last name

                                  select         @cnt          = count(*)
                                  from           customer (repeatableread)
                                  where          c_last        = @c_last and
                                                 c_w_id        = @c_w_id and
                                                 c_d_id        = @c_d_id
                                  select         @val = (@cnt + 1) / 2
                                  set            rowcount @val

                                  select         @c_id         = c_id
                                  from           customer (repeatableread)
                                  where          c_last        = @c_last and
                                                 c_w_id        = @c_w_id and
                                                 c_d_id        = @c_d_id
                                  order          by c_last, c_first

                                  set            rowcount 0
                 end

--  get customer info and update balances

                 update         customer
                 set            @c_balance = c_balance = c_balance - @h_amount,
                                c_payment_cnt          = c_payment_cnt + 1,
                                c_ytd_payment          = c_ytd_payment + @h_amount,
                                @c_first      = c_first,
                                @c_middle     = c_middle,
                                @c_last                = c_last,
                                @c_street_1   = c_street_1,
                                @c_street_2   = c_street_2,
                                @c_city                = c_city,
                                @c_state      = c_state,
                                @c_zip                 = c_zip,
                                @c_phone      = c_phone,
                                @c_credit     = c_credit,
                                @c_credit_lim          = c_credit_lim,
                                @c_discount            = c_discount,
                                @c_since      = c_since,
                                @data                  = c_data,
                                @c_id_local   = c_id
                 where          c_id                   = @c_id and
                                c_w_id                 = @c_w_id and
                                c_d_id                 = @c_d_id

--  if customer has bad credit get some more info

                 if (@c_credit = "BC")
                 begin

--               compute new info

                                  select @c_data         = convert(char(5),@c_id) +
                                                           convert(char(4),@c_d_id) +
                                                           convert(char(5),@c_w_id) +
                                                           convert(char(4),@d_id) +
                                                           convert(char(5),@w_id) +
                                                           convert(char(19),@h_amount) +
                                                           substring(@data, 1, 458)

--  update customer info
```

```
                update      customer
                set         c_data      = @c_data
                where       c_id        = @c_id and
                            c_w_id      = @c_w_id and
                            c_d_id      = @c_d_id

                select      @screen_data = substring (@c_data,1,200)
        end

-- get district data and update year-to-date

        update      district
        set         d_ytd                   = d_ytd + @h_amount,
                    @d_street_1= d_street_1,
                    @d_street_2= d_street_2,
                    @d_city                 = d_city,
                    @d_state    = d_state,
                    @d_zip                  = d_zip,
                    @d_name                 = d_name,
                    @d_id_local= d_id
        where       d_w_id                  = @w_id and
                    d_id                    = @d_id

-- get warehouse data and update year-to-date

        update      warehouse
        set         w_ytd                   = w_ytd + @h_amount,
                    @w_street_1             = w_street_1,
                    @w_street_2             = w_street_2,
                    @w_city                 = w_city,
                    @w_state    = w_state,
                    @w_zip                  = w_zip,
                    @w_name                 = w_name,
                    @w_id_local             = w_id
        where       w_id                    = @w_id

-- create history record

        insert into history values (@c_id_local,
                                                    @c_d_id,
                                                    @c_w_id,
                                                    @d_id_local,
                                                    @w_id_local,
                                                    @datetime,
                                                    @h_amount,
                                                    @w_name + "    " + @d_name)
commit tran p

-- return data to client

select      @c_id,
            @c_last,
            @datetime,
            @w_street_1,
            @w_street_2,
            @w_city,
            @w_state,
            @w_zip,
            @d_street_1,
            @d_street_2,
            @d_city,
            @d_state,
            @d_zip,
            @c_first,
            @c_middle,
            @c_street_1,
            @c_street_2,
            @c_city,
            @c_state,
            @c_zip,
            @c_phone,
            @c_since,
            @c_credit,
            @c_credit_lim,
            @c_discount,
            @c_balance,
            @screen_data

go
```

ordstat.sql

```
-- File:    ORDSTAT.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--          Copyright Microsoft, 1999
-- Purpose: Creates order status transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_orderstatus" )
            drop procedure   tpcc_orderstatus
go

create proc tpcc_orderstatus            @w_id       smallint,
                                        @d_id       tinyint,

                                        @c_id       int,
                                        @c_last     char(16) = ""

as

declare @c_balance                      numeric(12,2),
        @c_first                        char(16),
        @c_middle                       char(2),
        @o_id                           int,
        @o_entry_d                      datetime,
        @o_carrier_id                   smallint,
```

```
        @cnt                    smallint

begin tran o

if (@c_id = 0)
                begin

--  get customer id and info using last name

                        select  @cnt            = (count(*)+1)/2
                        from    customer (repeatableread)
                        where   c_last          = @c_last and
                                c_w_id          = @w_id and
                                c_d_id          = @d_id

                        set     rowcount @cnt

                        select  @c_id                   = c_id,
                                @c_balance = c_balance,
                                  @c_first    = c_first,
                                  @c_last     = c_last,
                                  @c_middle             = c_middle
                        from    customer (repeatableread)
                        where   c_last                  = @c_last and
                                c_w_id                  = @w_id and
                                c_d_id                  = @d_id
                        order   by c_w_id, c_d_id, c_last, c_first

                        set     rowcount 0
                end

                else

                begin

--  get customer info if by id

                        select  @c_balance = c_balance,
                                @c_first        = c_first,
                                @c_middle   = c_middle,
                                @c_last                 = c_last
                        from    customer (repeatableread)
                        where   c_id                    = @c_id and
                                c_d_id                  = @d_id and
                                c_w_id                  = @w_id

                        select  @cnt            = @@rowcount

                end

--  if no such customer

                if (@cnt = 0)
                begin
                        raiserror("Customer not found",18,1)
                        goto custnotfound
                end

--  get order info

                select  @o_id                   = o_id,
                        @o_entry_d = o_entry_d,
                        @o_carrier_id           = o_carrier_id
                from    orders (serializable)
                where   o_c_id                  = @c_id and
                        o_d_id                  = @d_id and
                        o_w_id                  = @w_id
                order   by o_id asc

--  select order lines for the current order

                select  ol_supply_w_id,
                        ol_i_id,
                        ol_quantity,
                        ol_amount,
                        ol_delivery_d
                from    order_line (repeatableread)
                where   ol_o_id = @o_id and
                        ol_d_id = @d_id and
                        ol_w_id = @w_id

custnotfound:

commit tran o

--  return data to client

select  @c_id,
        @c_last,
        @c_first,
        @c_middle,
        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id

go
```

delivery.sql

```
--  File:    DELIVERY.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--           Copyright Microsoft, 1999
--  Purpose:  Creates delivery transaction stored procedure
--
--           Interface Level: 4.10.000

use tpcc
go
```

```
if exists (select name from sysobjects where name = "tpcc_delivery" )
        drop procedure tpcc_delivery
go

create proc tpcc_delivery @w_id         smallint,
                                        @o_carrier_id   smallint
as

declare @d_id           tinyint,
    @o_id  int,
    @c_id  int,
    @total numeric(12,2),
    @oid1  int,
    @oid2  int,
    @oid3  int,
    @oid4  int,
    @oid5  int,
    @oid6  int,
    @oid7  int,
    @oid8  int,
    @oid9  int,
    @oid10          int

select @d_id = 0

begin tran d

        while (@d_id < 10)
        begin

            select  @d_id  = @d_id + 1,
                    @total = 0,
                    @o_id  = 0

                select      top 1
                            @o_id       = no_o_id
                from        new_order (serializable updlock)
                where       no_w_id     = @w_id and
                            no_d_id     = @d_id
                order       by no_o_id asc

            if (@@rowcount <> 0)
                begin

-- claim the order for this district

                            delete      new_order
                            where       no_w_id     = @w_id and
                                        no_d_id     = @d_id and
                                        no_o_id     = @o_id

-- set carrier_id on this order (and get customer id)

                            update      orders
                            set         o_carrier_id = @o_carrier_id,
                                        @c_id                   = o_c_id
                            where       o_w_id                  = @w_id and
                                        o_d_id                  = @d_id and
                                        o_id                    = @o_id

-- set date in all lineitems for this order (and sum amounts)

                            update      order_line
                            set         ol_delivery_d           = getdate(),
                                        @total                  = @total + ol_amount
                            where       ol_w_id                 = @w_id and
                                        ol_d_id                 = @d_id and
                                        ol_o_id                 = @o_id

-- accummulate lineitem amounts for this order into customer

                            update      customer
                            set         c_balance    = c_balance + @total,
                                        c_delivery_cnt          = c_delivery_cnt + 1
                            where       c_w_id                  = @w_id and
                                        c_d_id                  = @d_id and
                                        c_id                    = @c_id


                    end

                select @oid1 = case @d_id when  1  then @o_id else @oid1 end,
                @oid2 = case @d_id when  2  then @o_id else @oid2 end,
                @oid3 = case @d_id when  3  then @o_id else @oid3 end,
                @oid4 = case @d_id when  4  then @o_id else @oid4 end,
                @oid5 = case @d_id when  5  then @o_id else @oid5 end,
                @oid6 = case @d_id when  6  then @o_id else @oid6 end,
                @oid7 = case @d_id when  7  then @o_id else @oid7 end,
                @oid8 = case @d_id when  8  then @o_id else @oid8 end,
                @oid9 = case @d_id when  9  then @o_id else @oid9 end,
                @oid10 = case @d_id when 10 then @o_id else @oid10 end

        end

commit tran d

-- return delivery data to client

select @oid1,
    @oid2,
    @oid3,
    @oid4,
    @oid5,
    @oid6,
    @oid7,
    @oid8,
    @oid9,
    @oid10

go
```

stocklev.sql

```
-- File:    STOCKLEV.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--          Copyright Microsoft, 1999
-- Purpose: Creates stock level transaction stored procedure
--
--              Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_stocklevel" )
        drop procedure tpcc_stocklevel
go

create proc tpcc_stocklevel     @w_id       smallint,
                                @d_id       tinyint,
                                @threshhold     smallint
as

declare     @o_id_low int,
            @o_id_high int

select      @o_id_low  = (d_next_o_id - 20),
            @o_id_high = (d_next_o_id - 1)
from        district
where       d_w_id                  = @w_id and
            d_id                    = @d_id

select      count(distinct(s_i_id))
from        stock, order_line
where       ol_w_id                 =  @w_id and
            ol_d_id                 =  @d_id and
            ol_o_id                   between @o_id_low and
                                      @o_id_high and
            s_w_id                  =  ol_w_id and
            s_i_id                  =  ol_i_id and
            s_quantity    <  @threshhold

go
```

version.sql

```
-- File:    VERSION.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.20.000
--          Copyright Microsoft, 1999
-- Purpose: Returns version level of TPC-C stored procs
-- Note:    Always update the return value of this proc for
--          any interface changes or "must have" bug fixes.
--
-- The value returned by this SP defines the "interface level",
-- which must match between the stored procs and the client code.
-- The interface level may be down rev from the current kit.  This
-- indicates that the interface hasn't changed since that version.

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_version" )
        drop procedure tpcc_version
go

create proc tpcc_version
as
declare     @version    char(8)

begin
        select @version = "4.10.000"
        select @version as "Version"
end

go
```

Loader Source Code

tpcc.h

```
//          File:               TPCC.H
//                                  Microsoft TPC-C Kit Ver. 4.20
//                                  Copyright Microsoft, 1996, 1997, 1998, 1999
//          Purpose:    Header file for TPC-C database loader


// Build number of TPC Benchmark Kit
#define TPCKIT_VER    "4.20"

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <sys\types.h>

// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>
```

```
// General constants
#define MILLI                   1000
#define FALSE                   0
#define TRUE                    1
#define UNDEF                                   -1
#define MINPRINTASCII           32
#define MAXPRINTASCII           126

// Default environment constants
#define SERVER                  ""
#define DATABASE                "tpcc"
#define USER                    "sa"
#define PASSWORD                  ""

// Default loader arguments
#define BATCH                   10000
#define DEFLDPACKSIZE           32768
#define LOADER_RES_FILE          "logs\\load.out"
#define LOADER_NURAND_C          123
#define DEF_STARTING_WAREHOUSE   1
#define BUILD_INDEX                     1       // build both data and indexes
#define INDEX_ORDER                     1       // build indexes before load
#define SCALE_DOWN                      0  // build a normal scale database
#define INDEX_SCRIPT_PATH       "scripts"

typedef struct
{
   char                 *server;
   char                 *database;
   char                 *user;
   char                 *password;
        BOOL            tables_all;             // set if loading all
tables
        BOOL            table_item;             // set if loading
ITEM table specifically
        BOOL            table_warehouse;        // set if loading WAREHOUSE,
DISTRICT, and STOCK
        BOOL            table_customer;         // set if loading
CUSTOMER and HISTORY
        BOOL            table_orders;           // set if loading
NEW-ORDER, ORDERS, ORDER-LINE
   long                 num_warehouses;
   long                 batch;
   long                 verbose;
        long            pack_size;
        char            *loader_res_file;
        char            *synch_servername;
        long            case_sensitivity;
        long            starting_warehouse;
        long            build_index;
        long            index_order;
        long            scale_down;
        char            *index_script_path;
} TPCCLDR_ARGS;
```

```
// String length constants
        #define SERVER_NAME_LEN         20
        #define DATABASE_NAME_LEN       20
        #define USER_NAME_LEN           20
        #define PASSWORD_LEN            20
        #define TABLE_NAME_LEN          20
        #define I_DATA_LEN              50
        #define I_NAME_LEN              24
        #define BRAND_LEN               1
        #define LAST_NAME_LEN           16
        #define W_NAME_LEN              10
        #define ADDRESS_LEN             20
        #define STATE_LEN               2
        #define ZIP_LEN                 9
        #define S_DIST_LEN              24
        #define S_DATA_LEN              50
        #define D_NAME_LEN              10
        #define FIRST_NAME_LEN          16
        #define MIDDLE_NAME_LEN         2
        #define PHONE_LEN               16
        #define CREDIT_LEN              2
        #define C_DATA_LEN              500
        #define H_DATA_LEN              24
        #define DIST_INFO_LEN           24
        #define MAX_OL_NEW_ORDER_ITEMS  15
        #define MAX_OL_ORDER_STATUS_ITEMS 15
        #define STATUS_LEN              25
        #define OL_DIST_INFO_LEN                        24
        #define C_SINCE_LEN                                     23
        #define H_DATE_LEN                                      23
        #define OL_DELIVERY_D_LEN               23
        #define O_ENTRY_D_LEN           23


// Functions in random.c
void        seed();
long        irand();
double      drand();
void        WUCreate();
short       WURand();
long        RandomNumber(long lower, long upper);

// Functions in getargs.c;
void        GetArgsLoader();
void        GetArgsLoaderUsage();

// Functions in time.c
long        TimeNow();

// Functions in strings.c
void        MakeAddress();
void        LastName();
int         MakeAlphaString();
int         MakeOriginalAlphaString();
int         MakeNumberString();
int         MakeZipNumberString();
```

```
void        InitString();
void        InitAddress();
void        PaddString();
```

tpccldr.c

```
//          File:                   TPCCLDR.C
//                                          Microsoft TPC-C Kit Ver. 4.20
//                                          Copyright Microsoft, 1996, 1997, 1998, 1999
//          Purpose:    Source file for TPC-C database loader


// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS            100000
#define MAXITEMS_SCALE_DOWN                     100
#define CUSTOMERS_PER_DISTRICT  3000
#define CUSTOMERS_SCALE_DOWN            30
#define DISTRICT_PER_WAREHOUSE  10
#define ORDERS_PER_DISTRICT     3000
#define ORDERS_SCALE_DOWN       30
#define MAX_CUSTOMER_THREADS    2
#define MAX_ORDER_THREADS       3
#define MAX_MAIN_THREADS        4

// Functions declarations

void HandleErrorDBC (SQLHDBC  hdbc1);

void CheckSQL();
void CheckDataBase();

long NURand();
void LoadItem();
void LoadWarehouse();

void Stock();
void District();

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();

void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures

typedef struct
{
   long             ol;
   long             ol_i_id;
   short            ol_supply_w_id;
   short            ol_quantity;
   double            ol_amount;
   char             ol_dist_info[DIST_INFO_LEN+1];
            char                                    ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
   long             o_id;
   short            o_d_id;
   short            o_w_id;
   long             o_c_id;
   short            o_carrier_id;
   short            o_ol_cnt;
   short            o_all_local;
            ORDER_LINE_STRUCT   o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
   long                                     c_id;
   short                                    c_d_id;
   short                                    c_w_id;
            char                                    c_first[FIRST_NAME_LEN+1];
   char                                     c_middle[MIDDLE_NAME_LEN+1];
   char                                     c_last[LAST_NAME_LEN+1];
   char                                     c_street_1[ADDRESS_LEN+1];
   char                                     c_street_2[ADDRESS_LEN+1];
   char                                     c_city[ADDRESS_LEN+1];
   char                                     c_state[STATE_LEN+1];
   char                                     c_zip[ZIP_LEN+1];
   char                                     c_phone[PHONE_LEN+1];
   char                                     c_credit[CREDIT_LEN+1];
   double                                   c_credit_lim;
   double                                   c_discount;
//  fix to avoid ODBC float to numeric conversion problem.
//  double                                   c_balance;
            char                                    c_balance[6];

   double                                   c_ytd_payment;
            short                                   c_payment_cnt;
            short                                   c_delivery_cnt;
            char                                    c_data[C_DATA_LEN+1];
```

```
        double                  h_amount;
    char                        h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
        char                    c_last[LAST_NAME_LEN+1];
        char                    c_first[FIRST_NAME_LEN+1];
    long                        c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
    long        time_start;
} LOADER_TIME_STRUCT;


// Global variables

char        szLastError[300];

HENV        henv;

HDBC        v_hdbc;                                 // for SQL Server version
verification
HDBC        i_hdbc1;                                // for ITEM table
HDBC        w_hdbc1;                                // for WAREHOUSE, DISTRICT, STOCK
HDBC        c_hdbc1;                                // for CUSTOMER
HDBC        c_hdbc2;                                // for HISTORY
HDBC        o_hdbc1;                                // for ORDERS
HDBC        o_hdbc2;                                // for NEW-ORDER

HDBC        o_hdbc3;                                // for ORDER-LINE

HSTMT       v_hstmt;                                // for SQL Server version verification
HSTMT       i_hstmt1;
HSTMT       w_hstmt1;
HSTMT       c_hstmt1, c_hstmt2;
HSTMT       o_hstmt1, o_hstmt2, o_hstmt3;

ORDERS_STRUCT   orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long        orders_rows_loaded;
long        new_order_rows_loaded;
long        order_line_rows_loaded;
long        history_rows_loaded;
long        customer_rows_loaded;
long        stock_rows_loaded;
long        district_rows_loaded;
long        item_rows_loaded;
long        warehouse_rows_loaded;
long        main_time_start;
long        main_time_end;
long                    max_items;
long                    customers_per_district;
long                    orders_per_district;
long                    first_new_order;
long                    last_new_order;


TPCCLDR_ARGS    *aptr, args;


//==============================================================================
//
// Function name: main
//
//==============================================================================

int main(int  argc, char **argv)
{

        DWORD       dwThreadID[MAX_MAIN_THREADS];
        HANDLE      hThread[MAX_MAIN_THREADS];
        FILE        *fLoader;
        char        buffer[255];
        int                                         i;

        for (i=0; i<MAX_MAIN_THREADS; i++)
                hThread[i] = NULL;

  printf("\n*************************************************");
        printf("\n*                                               *");
        printf("\n* Microsoft SQL Server                          *");
        printf("\n*                                               *");
        printf("\n* TPC-C BENCHMARK KIT:  Database loader         *");
        printf("\n* Version %s                                    *",  TPCKIT_VER);
        printf("\n*                                               *");
        printf("\n*************************************************\n\n");

  // process command line arguments

  aptr = &args;
  GetArgsLoader(argc, argv, aptr);

        // verify correct SQL Server version in use
        // you must be using SQL Server 7.00.623 or better to load

        CheckSQL();

        // verify database and tables exist before attempting to load

        CheckDataBase();

        printf("Build interface is ODBC.\n");

        if (aptr->build_index == 0)
                printf("Data load only - no index creation.\n");
        else
                printf("Data load and index creation.\n");

        if (aptr->index_order == 0)
```

```
                printf("Clustered indexes will be created after bulk load.\n");
        else
                printf("Clustered indexes will be created before bulk load.\n");

        // set database scale values
        if (aptr->scale_down == 1)
        {
                printf("*** Scaled Down Database ***\n");
                max_items = MAXITEMS_SCALE_DOWN;
                customers_per_district = CUSTOMERS_SCALE_DOWN;
                orders_per_district = ORDERS_SCALE_DOWN;
                first_new_order = 0;
                last_new_order  = 30;
        }
        else
        {
                max_items = MAXITEMS;
                customers_per_district = CUSTOMERS_PER_DISTRICT;
                orders_per_district = ORDERS_PER_DISTRICT;
                first_new_order = 2100;
                last_new_order  = 3000;
        }


        // open connections to SQL Server

        OpenConnections();

        // open file for loader results
fLoader = fopen(aptr->loader_res_file, "w");

if (fLoader == NULL)
        {
                printf("Error, loader result file open failed.");
                exit(-1);
        }

// start loading data

sprintf(buffer,"TPC-C load started for %ld warehouses.\n",aptr->num_warehouses);

        printf("%s",buffer);
        fprintf(fLoader,"%s",buffer);

        main_time_start = (TimeNow() / MILLI);

        // start parallel load threads


        if (aptr->tables_all || aptr->table_item)
        {
                fprintf(fLoader, "\nStarting loader threads for: item\n");

                hThread[0] = CreateThread(NULL,
                                                                        0,
(LPTHREAD_START_ROUTINE) LoadItem,
                                                                        NULL,
                                                                        0,
                                                                        &dwThreadID[0]);


                if (hThread[0] == NULL)
                {
                        printf("Error, failed in creating creating thread = 0.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_warehouse)
        {
                fprintf(fLoader, "Starting loader threads for: warehouse\n");

                hThread[1] = CreateThread(NULL,
                                                                        0,
(LPTHREAD_START_ROUTINE) LoadWarehouse,
                                                                        NULL,
                                                                        0,
                                                                        &dwThreadID[1]);


                if (hThread[1] == NULL)
                {
                        printf("Error, failed in creating creating thread = 1.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_customer)
        {
                fprintf(fLoader, "Starting loader threads for: customer\n");

                hThread[2] = CreateThread(NULL,
                                                                        0,
(LPTHREAD_START_ROUTINE) LoadCustomer,
                                                                        NULL,
                                                                        0,
                                                                        &dwThreadID[2]);


                if (hThread[2] == NULL)
                {
                        printf("Error, failed in creating creating main thread = 2.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_orders)
        {
```

```
                fprintf(fLoader, "Starting loader threads for: orders\n");

                hThread[3] = CreateThread(NULL,
                                                                        0,

(LPTHREAD_START_ROUTINE) LoadOrders,
                                                                        NULL,
                                                                        0,
                                                                        &dwThreadID[3]);


                if (hThread[3] == NULL)
                {
                        printf("Error, failed in creating creating main thread = 3.\n");
                        exit(-1);
                }
        }

        // Wait for threads to finish...
        for (i=0; i<MAX_MAIN_THREADS; i++)
        {
                if (hThread[i] != NULL)
                {
                        WaitForSingleObject( hThread[i], INFINITE );
                        CloseHandle(hThread[i]);
                        hThread[i] = NULL;
                }
        }

        main_time_end = (TimeNow() / MILLI);

   sprintf(buffer,"\nTPC-C load completed successfully in %ld minutes.\n",
                        (main_time_end - main_time_start)/60);

        printf("%s",buffer);
        fprintf(fLoader,"%s", buffer);

        fclose(fLoader);

        SQLFreeEnv(henv);

   exit(0);

        return 0;
}


//==============================================================
//
// Function name: LoadItem
//
//==============================================================

void LoadItem()
{
   long                 i_id;
        long                    i_im_id;
char            i_name[I_NAME_LEN+1];
double          i_price;
char            i_data[I_DATA_LEN+1];
        char                    name[20];
        long                    time_start;
        RETCODE                 rc;
        DBINT                   rcint;
        char                    bcphint[128];

        // Seed with unique number
        seed(1);

        printf("Loading item table...\n");

        // if build index before load
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxitmcl");

        InitString(i_name,I_NAME_LEN+1);
        InitString(i_data,I_DATA_LEN+1);

        sprintf(name, "%s..%s", aptr->database, "item");

        rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (i_id), ROWS_PER_BATCH = 100000");
                rc = bcp_control(i_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);
        }

        rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL, 0, 0, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL, 0, 0, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);
```

```
        time_start = (TimeNow() / MILLI);

        item_rows_loaded = 0;

        for (i_id = 1; i_id <= max_items; i_id++)
        {
                i_im_id = RandomNumber(1L, 10000L);

                MakeAlphaString(14, 24, I_NAME_LEN, i_name);

                i_price = ((float) RandomNumber(100L, 10000L))/100.0;

                MakeOriginalAlphaString(26,50, I_DATA_LEN, i_data, 10);

                rc = bcp_sendrow(i_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                item_rows_loaded++;
                CheckForCommit(i_hdbc1,i_hstmt1, item_rows_loaded, "item", &time_start);
        }

        rcint = bcp_done(i_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(i_hdbc1);

        printf("Finished loading item table.\n");

        SQLFreeStmt(i_hstmt1,SQL_DROP);
        SQLDisconnect(i_hdbc1);
        SQLFreeConnect(i_hdbc1);

        // if build index after load
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxitmcl");

}


//============================================================================
//
// Function   : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are created
//
//============================================================================

void LoadWarehouse()
{
   short        w_id;
   char         w_name[W_NAME_LEN+1];
   char         w_street_1[ADDRESS_LEN+1];
   char         w_street_2[ADDRESS_LEN+1];
   char         w_city[ADDRESS_LEN+1];
   char         w_state[STATE_LEN+1];
   char         w_zip[ZIP_LEN+1];
   double       w_tax;
   double       w_ytd;
                char            name[20];
                long            time_start;
                RETCODE         rc;
                DBINT           rcint;
                char            bcphint[128];


        // Seed with unique number
        seed(2);

        printf("Loading warehouse table...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxwarcl");

        InitString(w_name,W_NAME_LEN+1);
        InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

        sprintf(name, "%s..%s", aptr->database, "warehouse");

        rc = bcp_init(w_hdbc1, name, NULL, "logs\\whouse.err", DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (w_id), ROWS_PER_BATCH = %d", aptr->num_warehouses);
                rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);
        }

        rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN, NULL, 0, 0, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN, NULL, 0, 0, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN, NULL, 0, 0, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL, 0, 0, 6);
```

```
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0, 7);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 8);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	time_start = (TimeNow() / MILLI);

	warehouse_rows_loaded = 0;

	for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
	{
		MakeAlphaString(6,10, W_NAME_LEN, w_name);

		MakeAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

		w_tax = ((float) RandomNumber(0L,2000L))/10000.00;

		w_ytd = 300000.00;

		rc = bcp_sendrow(w_hdbc1);
		if (rc != SUCCEED)
			HandleErrorDBC(w_hdbc1);

		warehouse_rows_loaded++;
		CheckForCommit(w_hdbc1, i_hstmt1, warehouse_rows_loaded, "warehouse", &time_start);
	}

	rcint = bcp_done(w_hdbc1);
	if (rcint < 0)
		HandleErrorDBC(w_hdbc1);

	printf("Finished loading warehouse table.\n");

	// if build index after load...
	if ((aptr->build_index == 1) && (aptr->index_order == 0))
		BuildIndex("idxwarcl");

	stock_rows_loaded = 0;
	district_rows_loaded = 0;

	District();
	Stock();

}
```

```
//==============================================================
//
// Function   : District
//
//==============================================================

void District()
{
   short	d_id;
   short	d_w_id;
   char		d_name[D_NAME_LEN+1];
   char		d_street_1[ADDRESS_LEN+1];
   char		d_street_2[ADDRESS_LEN+1];
   char		d_city[ADDRESS_LEN+1];
   char		d_state[STATE_LEN+1];
   char		d_zip[ZIP_LEN+1];
   double	d_tax;
   double	d_ytd;
		char		name[20];
   long		d_next_o_id;
		long		time_start;
		int				w_id;
		RETCODE	rc;
		DBINT		rcint;
		char		bcphint[128];

		// Seed with unique number
		seed(4);

		printf("Loading district table...\n");

		// build index before load
		if ((aptr->build_index == 1) && (aptr->index_order == 1))
			BuildIndex("idxdiscl");

		InitString(d_name, D_NAME_LEN+1);
		InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
		sprintf(name, "%s..%s", aptr->database, "district");

		rc = bcp_init(w_hdbc1, name, NULL, "logs\\district.err", DB_IN);
		if (rc != SUCCEED)
			HandleErrorDBC(w_hdbc1);

		if ((aptr->build_index == 1) && (aptr->index_order == 1))
		{
			sprintf(bcphint, "tablock, order (d_w_id, d_id), ROWS_PER_BATCH = %u",
(aptr->num_warehouses * 10));
			rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
			if (rc != SUCCEED)
				HandleErrorDBC(w_hdbc1);
		}

		rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
		if (rc != SUCCEED)
			HandleErrorDBC(w_hdbc1);
```

```
rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0, D_NAME_LEN, NULL, 0, 0, 3);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 4);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0, ADDRESS_LEN, NULL, 0, 0, 5);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0, ADDRESS_LEN, NULL, 0, 0, 6);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0, STATE_LEN, NULL, 0, 0, 7);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0, 0, 8);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 10);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 11);
if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

d_ytd  = 30000.0;

d_next_o_id = orders_per_district+1;

time_start = (TimeNow() / MILLI);

for (w_id = aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
{
        d_w_id = w_id;

        for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
        {
                MakeAlphaString(6,10,D_NAME_LEN, d_name);

                MakeAddress(d_street_1, d_street_2, d_city, d_state, d_zip);

                d_tax = ((float) RandomNumber(0L,2000L))/10000.00;

                rc = bcp_sendrow(w_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                district_rows_loaded++;
                CheckForCommit(w_hdbc1, w_hstmt1, district_rows_loaded, "district", &time_start);
        }
  }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

        printf("Finished loading district table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxdiscl");

  return;
}


//==============================================================
//
// Function   : Stock
//
//==============================================================

void Stock()
{
  long      s_i_id;
  short     s_w_id;
  short     s_quantity;
  char      s_dist_01[S_DIST_LEN+1];
  char      s_dist_02[S_DIST_LEN+1];
  char      s_dist_03[S_DIST_LEN+1];
  char      s_dist_04[S_DIST_LEN+1];
  char      s_dist_05[S_DIST_LEN+1];
  char      s_dist_06[S_DIST_LEN+1];
  char      s_dist_07[S_DIST_LEN+1];
  char      s_dist_08[S_DIST_LEN+1];
  char      s_dist_09[S_DIST_LEN+1];
  char      s_dist_10[S_DIST_LEN+1];
  long      s_ytd;
  short     s_order_cnt;
  short     s_remote_cnt;
  char      s_data[S_DATA_LEN+1];
  short     len;
        char        name[20];
        long        time_start;
        RETCODE     rc;
        DBINT       rcint;
        char        bcphint[128];
```

```
	// Seed with unique number
	seed(3);

	// if build index before load...
	if ((aptr->build_index == 1) && (aptr->index_order == 1))
		BuildIndex("idxstkcl");

	sprintf(name, "%s..%s", aptr->database, "stock");

	rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err", DB_IN);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	if ((aptr->build_index == 1) && (aptr->index_order == 1))
	{
		sprintf(bcphint, "tablock, order (s_i_id, s_w_id), ROWS_PER_BATCH = %u",
(aptr->num_warehouses * 100000));
		rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
		if (rc != SUCCEED)
			HandleErrorDBC(w_hdbc1);
	}

	rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN, NULL, 0, 0, 4);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN, NULL, 0, 0, 5);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN, NULL, 0, 0, 6);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN, NULL, 0, 0, 7);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN, NULL, 0, 0, 8);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN, NULL, 0, 0, 9);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN, NULL, 0, 0, 10);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN, NULL, 0, 0, 11);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN, NULL, 0, 0, 12);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN, NULL, 0, 0, 13);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 14);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 15);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 16);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL, 0, 0, 17);
	if (rc != SUCCEED)
		HandleErrorDBC(w_hdbc1);

	s_ytd = s_order_cnt = s_remote_cnt = 0;

	time_start = (TimeNow() / MILLI);

	printf("...Loading stock table\n");

	for (s_i_id=1; s_i_id <= max_items; s_i_id++)
	{

		for (s_w_id = (short)aptr->starting_warehouse; s_w_id <= aptr->num_warehouses; s_w_id++)
		{

			s_quantity = (short)RandomNumber(10L,100L);
			len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_01);
			len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_02);
			len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_03);
			len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_04);
			len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_05);
			len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_06);
			len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_07);
			len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_08);
```

```
                                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_09);
                                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_10);

                                len = MakeOriginalAlphaString(26,50,S_DATA_LEN, s_data,10);

                                rc = bcp_sendrow(w_hdbc1);
                                if (rc != SUCCEED)
                                        HandleErrorDBC(w_hdbc1);

                                stock_rows_loaded++;
                                CheckForCommit(w_hdbc1, w_hstmt1, stock_rows_loaded, "stock", &time_start);

                        }
                }

                rcint = bcp_done(w_hdbc1);
                if (rcint < 0)
                        HandleErrorDBC(w_hdbc1);

                printf("Finished loading stock table.\n");

                SQLFreeStmt(w_hstmt1, SQL_DROP);
                SQLDisconnect(w_hdbc1);
                SQLFreeConnect(w_hdbc1);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxstkcl");

                return;
}


//================================================================
//
// Function   : LoadCustomer
//
//================================================================

void LoadCustomer()
{
                LOADER_TIME_STRUCT      customer_time_start;
                LOADER_TIME_STRUCT      history_time_start;
                short                   w_id;
   short                d_id;
                DWORD                           dwThreadID[MAX_CUSTOMER_THREADS];
                HANDLE                          hThread[MAX_CUSTOMER_THREADS];
                char                    name[20];
                RETCODE                                         rc;
                DBINT                                           rcint;
                char                                            bcphint[128];
                char                                            cmd[256];
                // SQLRETURN                                    rc_1;
                // SQLSMALLINT                                  recnum, MsgLen;
                // SQLCHAR                                              SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
                // SQLINTEGER                                           NativeError;

                // Seed with unique number
                seed(5);

                printf("Loading customer and history tables...\n");

                // if build index before load...
                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                        BuildIndex("idxcuscl");

                // Initialize bulk copy
                sprintf(name, "%s..%s", aptr->database, "customer");

                rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err", DB_IN);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                        sprintf(bcphint, "tablock, order (c_w_id, c_d_id, c_id), ROWS_PER_BATCH = %u",
(aptr->num_warehouses * 30000));
                        rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                HandleErrorDBC(c_hdbc1);
                }

                sprintf(name, "%s..%s", aptr->database, "history");

                rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

                sprintf(bcphint, "tablock");
                rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

                customer_rows_loaded    = 0;
                history_rows_loaded     = 0;

                CustomerBufInit();

                customer_time_start.time_start = (TimeNow() / MILLI);
                history_time_start.time_start = (TimeNow() / MILLI);

                for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
                {
                        for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
                        {

                                CustomerBufLoad(d_id, w_id);

                                // Start parallel loading threads here...

                                // Start customer table thread
```

```
                printf("...Loading customer table for: d_id = %d, w_id = %d\n", d_id, w_id);

                hThread[0] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadCustomerTable,
&customer_time_start,
                                                                                0,
&dwThreadID[0]);
                if (hThread[0] == NULL)
                {
                        printf("Error, failed in creating creating thread = 0.\n");
                        exit(-1);
                }

                // Start History table thread

                printf("...Loading history table for: d_id = %d, w_id = %d\n", d_id, w_id);

                hThread[1] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadHistoryTable,
&history_time_start,
                                                                                0,
&dwThreadID[1]);
                if (hThread[1] == NULL)
                {
                        printf("Error, failed in creating creating thread = 1.\n");
                        exit(-1);
                }

                WaitForSingleObject( hThread[0], INFINITE );
                WaitForSingleObject( hThread[1], INFINITE );

                if (CloseHandle(hThread[0]) == FALSE)
                {
                        printf("Error, failed in closing customer thread handle witherrno: %d\n",
GetLastError());
                }

                if (CloseHandle(hThread[1]) == FALSE)
                {
                        printf("Error, failed in closing history thread handle witherrno: %d\n",
GetLastError());
                }

        }
```

```
        }

        // flush the bulk connection
        rcint = bcp_done(c_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(c_hdbc1);

        rcint = bcp_done(c_hdbc2);
        if (rcint < 0)
                HandleErrorDBC(c_hdbc2);

        printf("Finished loading customer table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxcuscl");

        // build non-clustered index
        if (aptr->build_index == 1)
                BuildIndex("idxcusnc");

        // Output the NURAND used for the loader into C_FIRST for C_ID = 1,
        // C_W_ID = 1, and C_D_ID = 1
        sprintf(cmd, "isql -S%s -U%s -P%s -d%s -e -Q\"update customer set c_first = 'C_LOAD = %d' where c_id = 1
and c_w_id = 1 and c_d_id = 1\" > logs\\nurand_load.log",
                                        aptr->server,
                                        aptr->user,
                                        aptr->password,
                                        aptr->database,
                                        LOADER_NURAND_C);

        system(cmd);

        SQLFreeStmt(c_hstmt1, SQL_DROP);
        SQLDisconnect(c_hdbc1);
        SQLFreeConnect(c_hdbc1);

        SQLFreeStmt(c_hstmt2, SQL_DROP);
        SQLDisconnect(c_hdbc2);
        SQLFreeConnect(c_hdbc2);


  return;

}


//============================================================================
//
// Function   : CustomerBufInit
//
//============================================================================

void CustomerBufInit()
{
```

```
	int	i;

	for (i=0;i<customers_per_district;i++)
	{
		customer_buf[i].c_id = 0;
		customer_buf[i].c_d_id = 0;
		customer_buf[i].c_w_id = 0;

		strcpy(customer_buf[i].c_first,"");
		strcpy(customer_buf[i].c_middle,"");
		strcpy(customer_buf[i].c_last,"");
		strcpy(customer_buf[i].c_street_1,"");
		strcpy(customer_buf[i].c_street_2,"");
		strcpy(customer_buf[i].c_city,"");
		strcpy(customer_buf[i].c_state,"");
		strcpy(customer_buf[i].c_zip,"");
		strcpy(customer_buf[i].c_phone,"");
		strcpy(customer_buf[i].c_credit,"");

		customer_buf[i].c_credit_lim= 0;
		customer_buf[i].c_discount= (float) 0;

		//  fix to avoid ODBC float to numeric conversion problem.
		//		customer_buf[i].c_balance= 0;
		strcpy(customer_buf[i].c_balance,"");

		customer_buf[i].c_ytd_payment= 0;
		customer_buf[i].c_payment_cnt= 0;
		customer_buf[i].c_delivery_cnt= 0;

		strcpy(customer_buf[i].c_data,"");

		customer_buf[i].h_amount= 0;

		strcpy(customer_buf[i].h_data,"");

	}


}


//================================================================
//
// Function   : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//================================================================

void CustomerBufLoad(int d_id, int w_id)
{
	long                                                    i;
  CUSTOMER_SORT_STRUCT    c[CUSTOMERS_PER_DISTRICT];


	for (i=0;i<customers_per_district;i++)
{
	if (i < 1000)
		LastName(i, c[i].c_last);
	else
		LastName(NURand(255,0,999,LOADER_NURAND_C), c[i].c_last);

	MakeAlphaString(8,16,FIRST_NAME_LEN, c[i].c_first);

	c[i].c_id = i+1;

}

printf("...Loading customer buffer for: d_id = %d, w_id = %d\n",
		d_id, w_id);

for (i=0;i<customers_per_district;i++)
{

	customer_buf[i].c_d_id = d_id;
	customer_buf[i].c_w_id = w_id;
	customer_buf[i].h_amount= 10.0;

	customer_buf[i].c_ytd_payment = 10.0;

	customer_buf[i].c_payment_cnt = 1;
	customer_buf[i].c_delivery_cnt= 0;

	// Generate CUSTOMER and HISTORY data

	customer_buf[i].c_id = c[i].c_id;

	strcpy(customer_buf[i].c_first, c[i].c_first);
	strcpy(customer_buf[i].c_last, c[i].c_last);

	customer_buf[i].c_middle[0]= 'O';
	customer_buf[i].c_middle[1]= 'E';

	MakeAddress(customer_buf[i].c_street_1,
				customer_buf[i].c_street_2,
				customer_buf[i].c_city,
				customer_buf[i].c_state,
				customer_buf[i].c_zip);

	MakeNumberString(16, 16, PHONE_LEN, customer_buf[i].c_phone);

	if (RandomNumber(1L, 100L) > 10)
		customer_buf[i].c_credit[0]= 'G';
	else
		customer_buf[i].c_credit[0]= 'B';
	customer_buf[i].c_credit[1]= 'C';

	customer_buf[i].c_credit_lim= 50000.0;
	customer_buf[i].c_discount= ((float) RandomNumber(0L, 5000L)) / 10000.0;


	//  fix to avoid ODBC float to numeric conversion problem.
```

```
                        //  customer_buf[i].c_balance = -10.0;
                        strcpy(customer_buf[i].c_balance,"-10.0");

                        MakeAlphaString(300, 500, C_DATA_LEN, customer_buf[i].c_data);

                        // Generate HISTORY data
                        MakeAlphaString(12, 24, H_DATA_LEN, customer_buf[i].h_data);

                }
}

//===================================================================
//
// Function   : LoadCustomerTable
//
//===================================================================

void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
                int     i;
    long        c_id;
    short       c_d_id;
    short       c_w_id;
    char        c_first[FIRST_NAME_LEN+1];
    char        c_middle[MIDDLE_NAME_LEN+1];
    char        c_last[LAST_NAME_LEN+1];
    char        c_street_1[ADDRESS_LEN+1];
    char        c_street_2[ADDRESS_LEN+1];
    char        c_city[ADDRESS_LEN+1];
    char        c_state[STATE_LEN+1];
    char        c_zip[ZIP_LEN+1];
    char        c_phone[PHONE_LEN+1];
    char        c_credit[CREDIT_LEN+1];
    double      c_credit_lim;
    double      c_discount;

                //  fix to avoid ODBC float to numeric conversion problem.
                //  double                  c_balance;
                char                    c_balance[6];

    double      c_ytd_payment;
    short       c_payment_cnt;
    short       c_delivery_cnt;
    char        c_data[C_DATA_LEN+1];
                char                    c_since[C_SINCE_LEN+1];
                RETCODE                 rc;

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

                rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 0, 4);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN,NULL, 0, 0, 5);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0, 6);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 7);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL,0,0, 8);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, 12);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

                rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0, SQLCHARACTER, 13);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, 14);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 15);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 16);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);
```

```
		//  fix to avoid ODBC float to numeric conversion problem.
//  rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 17);
		//  if (rc != SUCCEED)
		//		HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0, SQLCHARACTER, 17);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);


rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 18);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 19);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt,0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 20);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500, NULL, 0, 0, 21);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);

for (i = 0; i < customers_per_district; i++)
{
			c_id = customer_buf[i].c_id;
			c_d_id = customer_buf[i].c_d_id;
			c_w_id = customer_buf[i].c_w_id;

			strcpy(c_first, customer_buf[i].c_first);
			strcpy(c_middle, customer_buf[i].c_middle);
			strcpy(c_last, customer_buf[i].c_last);
			strcpy(c_street_1, customer_buf[i].c_street_1);
			strcpy(c_street_2, customer_buf[i].c_street_2);
			strcpy(c_city, customer_buf[i].c_city);
			strcpy(c_state, customer_buf[i].c_state);
			strcpy(c_zip, customer_buf[i].c_zip);
			strcpy(c_phone, customer_buf[i].c_phone);
			strcpy(c_credit, customer_buf[i].c_credit);

			FormatDate(&c_since);

			c_credit_lim = customer_buf[i].c_credit_lim;
			c_discount = customer_buf[i].c_discount;

			//  fix to avoid ODBC float to numeric conversion problem.
			//  c_balance = customer_buf[i].c_balance;
			strcpy(c_balance, customer_buf[i].c_balance);

			c_ytd_payment = customer_buf[i].c_ytd_payment;
			c_payment_cnt = customer_buf[i].c_payment_cnt;
			c_delivery_cnt = customer_buf[i].c_delivery_cnt;

			strcpy(c_data, customer_buf[i].c_data);

			// Send data to server
			rc = bcp_sendrow(c_hdbc1);
			if (rc != SUCCEED)
				HandleErrorDBC(c_hdbc1);

			customer_rows_loaded++;
			CheckForCommit(c_hdbc1, c_hstmt1, customer_rows_loaded, "customer",
&customer_time_start->time_start);
   }

}


//====================================================================
//
// Function   : LoadHistoryTable
//
//====================================================================

void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
		int	i;
   long	c_id;
   short	c_d_id;
   short	c_w_id;
		double	h_amount;
   char	h_data[H_DATA_LEN+1];
		char			h_date[H_DATE_LEN+1];
		RETCODE		rc;

   rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 4);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 5);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

		rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN, NULL, 0, SQLCHARACTER, 6);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

   rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 7);
```

```
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

	rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, 8);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc2);

	for (i = 0; i < customers_per_district; i++)
	{
			c_id = customer_buf[i].c_id;
			c_d_id = customer_buf[i].c_d_id;
			c_w_id = customer_buf[i].c_w_id;
			h_amount = customer_buf[i].h_amount;
			strcpy(h_data, customer_buf[i].h_data);

			FormatDate(&h_date);

			// send to server
			rc = bcp_sendrow(c_hdbc2);
			if (rc != SUCCEED)
				HandleErrorDBC(c_hdbc2);

			history_rows_loaded++;
			CheckForCommit(c_hdbc2, c_hstmt2, history_rows_loaded, "history",
&history_time_start->time_start);
	}

}


//============================================================-==============
//
// Function   : LoadOrders
//
//============================================================================

void LoadOrders()
{
		LOADER_TIME_STRUCT	orders_time_start;
		LOADER_TIME_STRUCT	new_order_time_start;
		LOADER_TIME_STRUCT	order_line_time_start;
		short			w_id;
	short			d_id;
		DWORD				dwThreadID[MAX_ORDER_THREADS];
		HANDLE				hThread[MAX_ORDER_THREADS];
		char			name[20];
		RETCODE						rc;
		char						bcphint[128];

		// seed with unique number
		seed(6);

		printf("Loading orders...\n");

		// if build index before load...
		if ((aptr->build_index == 1) && (aptr->index_order == 1))
		{
				BuildIndex("idxordcl");
				BuildIndex("idxnodcl");
				BuildIndex("idxodlcl");
		}

		// initialize bulk copy
		sprintf(name, "%s..%s", aptr->database, "orders");

		rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);

		if ((aptr->build_index == 1) && (aptr->index_order == 1))
		{
			sprintf(bcphint, "tablock, order (o_w_id, o_d_id, o_id), ROWS_PER_BATCH = %u",
(aptr->num_warehouses * 30000));
			rc = bcp_control(o_hdbc1, BCPHINTS, (void*) bcphint);
			if (rc != SUCCEED)
				HandleErrorDBC(o_hdbc1);
		}

		sprintf(name, "%s..%s", aptr->database, "new_order");

		rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err", DB_IN);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc2);

		if ((aptr->build_index == 1) && (aptr->index_order == 1))
		{
			sprintf(bcphint, "tablock, order (no_w_id, no_d_id, no_o_id), ROWS_PER_BATCH = %u",
(aptr->num_warehouses * 9000));
			rc = bcp_control(o_hdbc2, BCPHINTS, (void*) bcphint);
			if (rc != SUCCEED)
				HandleErrorDBC(o_hdbc2);
		}

		sprintf(name, "%s..%s", aptr->database, "order_line");

		rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err", DB_IN);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc3);

	if ((aptr->build_index == 1) && (aptr->index_order == 1))
		{
			sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id, ol_o_id, ol_number), ROWS_PER_BATCH =
%u", (aptr->num_warehouses * 300000));
			rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcphint);
			if (rc != SUCCEED)
				HandleErrorDBC(o_hdbc3);
		}

		orders_rows_loaded	= 0;
		new_order_rows_loaded	= 0;
		order_line_rows_loaded	= 0;
```

```
        OrdersBufInit();

        orders_time_start.time_start = (TimeNow() / MILLI);
        new_order_time_start.time_start = (TimeNow() / MILLI);
        order_line_time_start.time_start = (TimeNow() / MILLI);

        for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
        {
                for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
                {

                        OrdersBufLoad(d_id, w_id);

                        // start parallel loading threads here...

                        // start Orders table thread

                        printf("...Loading Order Table for: d_id = %d, w_id = %d\n", d_id, w_id);

                        hThread[0] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadOrdersTable,
&orders_time_start,
                                                                                0,
&dwThreadID[0]);

                        if (hThread[0] == NULL)
                        {
                                printf("Error, failed in creating creating thread = 0.\n");
                                exit(-1);
                        }

                        // start NewOrder table thread

                        printf("...Loading New-Order Table for: d_id = %d, w_id = %d\n", d_id, w_id);

                        hThread[1] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadNewOrderTable,
&new_order_time_start,
                                                                                0,
&dwThreadID[1]);

                        if (hThread[1] == NULL)
                        {
                                printf("Error, failed in creating creating thread = 1.\n");
                                exit(-1);
                        }

                        // start Order-Line table thread

                        printf("...Loading Order-Line Table for: d_id = %d, w_id = %d\n", d_id, w_id);

                        hThread[2] = CreateThread(NULL,
                                                                                0,
(LPTHREAD_START_ROUTINE) LoadOrderLineTable,
&order_line_time_start,
                                                                                0,
&dwThreadID[2]);

                        if (hThread[2] == NULL)
                        {
                                printf("Error, failed in creating creating thread = 2.\n");
                                exit(-1);
                        }

                        WaitForSingleObject( hThread[0], INFINITE );
                        WaitForSingleObject( hThread[1], INFINITE );
                        WaitForSingleObject( hThread[2], INFINITE );

                        if (CloseHandle(hThread[0]) == FALSE)
                        {
                                printf("Error, failed in closing Orders thread handle with errno: %d\n",
GetLastError());
                        }

                        if (CloseHandle(hThread[1]) == FALSE)
                        {
                                printf("Error, failed in closing NewOrder thread handle with errno:
%d\n", GetLastError());
                        }

                        if (CloseHandle(hThread[2]) == FALSE)
                        {
                                printf("Error, failed in closing OrderLine thread handle with errno:
%d\n", GetLastError());
                        }

                }
        }

        printf("Finished loading orders.\n");

    return;
}


//==============================================================
//
// Function   : OrdersBufInit
//
// Clears shared buffer for ORDERS, NEWORDER, and ORDERLINE
```

```
//
//=====================================================================

void OrdersBufInit()
{
	int	i;
	int				j;

	for (i=0;i<orders_per_district;i++)
	{
		orders_buf[i].o_id = 0;
		orders_buf[i].o_d_id = 0;
		orders_buf[i].o_w_id = 0;
		orders_buf[i].o_c_id = 0;
		orders_buf[i].o_carrier_id = 0;
		orders_buf[i].o_ol_cnt = 0;
		orders_buf[i].o_all_local = 0;

		for (j=0;j<=14;j++)
		{
			orders_buf[i].o_ol[j].ol = 0;
			orders_buf[i].o_ol[j].ol_i_id = 0;
			orders_buf[i].o_ol[j].ol_supply_w_id = 0;
			orders_buf[i].o_ol[j].ol_quantity = 0;
			orders_buf[i].o_ol[j].ol_amount = 0;
			strcpy(orders_buf[i].o_ol[j].ol_dist_info,"");
		}
	}


}


//=====================================================================
//
// Function   : OrdersBufLoad
//
// Fills shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//=====================================================================

void OrdersBufLoad(int d_id, int w_id)
{

	int	cust[ORDERS_PER_DISTRICT+1];
	long	o_id;
	short	ol;

	printf("...Loading Order Buffer for: d_id = %d, w_id = %d\n",
				d_id, w_id);

	GetPermutation(cust, orders_per_district);

	for (o_id=0;o_id<orders_per_district;o_id++)
	{

		// Generate ORDER and NEW-ORDER data

		orders_buf[o_id].o_d_id = d_id;
		orders_buf[o_id].o_w_id = w_id;
		orders_buf[o_id].o_id = o_id+1;
		orders_buf[o_id].o_c_id = cust[o_id+1];
		orders_buf[o_id].o_ol_cnt = (short)RandomNumber(5L, 15L);

		if (o_id < first_new_order)
		{
			orders_buf[o_id].o_carrier_id = (short)RandomNumber(1L, 10L);
			orders_buf[o_id].o_all_local = 1;
		}
		else
		{
			orders_buf[o_id].o_carrier_id = 0;
			orders_buf[o_id].o_all_local = 1;
		}

		for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
		{

			orders_buf[o_id].o_ol[ol].ol = ol+1;
			orders_buf[o_id].o_ol[ol].ol_i_id = RandomNumber(1L, max_items);
			orders_buf[o_id].o_ol[ol].ol_supply_w_id = w_id;
			orders_buf[o_id].o_ol[ol].ol_quantity = 5;
			MakeAlphaString(24, 24, OL_DIST_INFO_LEN,
&orders_buf[o_id].o_ol[ol].ol_dist_info);

			// Generate ORDER-LINE data
			if (o_id < first_new_order)
			{
				orders_buf[o_id].o_ol[ol].ol_amount = 0;
				// Added to insure ol_delivery_d set properly during load

				FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);

			}
			else
			{
				orders_buf[o_id].o_ol[ol].ol_amount =
RandomNumber(1,999999)/100.0;
				// Added to insure ol_delivery_d set properly during load


				// odbc datetime format
				strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-12-31
00:00:00.000");

			}
		}
	}
}

//=====================================================================
```

```
//
// Function   : LoadOrdersTable
//
//============================================================================

void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
        int     i;
    long    o_id;
        short       o_d_id;
        short       o_w_id;
    long    o_c_id;
    short   o_carrier_id;
    short   o_ol_cnt;
    short   o_all_local;
        char                    o_entry_d[O_ENTRY_D_LEN+1];
        RETCODE                 rc;
        DBINT                   rcint;

        // bind ORDER data
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0, O_ENTRY_D_LEN, NULL, 0, SQLCHARACTER, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    for (i = 0; i < orders_per_district; i++)
    {
                o_id        = orders_buf[i].o_id;
                o_d_id      = orders_buf[i].o_d_id;
                o_w_id      = orders_buf[i].o_w_id;
                o_c_id      = orders_buf[i].o_c_id;
                o_carrier_id = orders_buf[i].o_carrier_id;
                o_ol_cnt    = orders_buf[i].o_ol_cnt;
                o_all_local = orders_buf[i].o_all_local;

                FormatDate(&o_entry_d);

                // send data to server
                rc = bcp_sendrow(o_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

                orders_rows_loaded++;
                CheckForCommit(o_hdbc1, o_hstmt1, orders_rows_loaded, "orders",
&orders_time_start->time_start);
    }

        // rcint = bcp_batch(o_hdbc1);
        // if (rcint < 0)
        //      HandleErrorDBC(o_hdbc1);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc1);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc1);

                SQLFreeStmt(o_hstmt1, SQL_DROP);
                SQLDisconnect(o_hdbc1);
                SQLFreeConnect(o_hdbc1);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxordcl");

                // build non-clustered index
                if (aptr->build_index == 1)
                        BuildIndex("idxordnc");
        }

}


//============================================================================
//
// Function   : LoadNewOrderTable
//
//============================================================================

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    int     i;
    long    o_id;
    short   o_d_id;
    short   o_w_id;
```

```
        RETCODE         rc;
        DBINT           rcint;

        // Bind NEW-ORDER data

rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        for (i = first_new_order; i < last_new_order; i++)
        {
                o_id    = orders_buf[i].o_id;
                o_d_id  = orders_buf[i].o_d_id;
                o_w_id  = orders_buf[i].o_w_id;

                rc = bcp_sendrow(o_hdbc2);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);

                new_order_rows_loaded++;
                CheckForCommit(o_hdbc2, o_hstmt2, new_order_rows_loaded, "new_order",
&new_order_time_start->time_start);
        }

        // rcint = bcp_batch(o_hdbc2);
        // if (rcint < 0)
        //      HandleErrorDBC(o_hdbc2);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc2);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc2);

                SQLFreeStmt(o_hstmt2, SQL_DROP);
                SQLDisconnect(o_hdbc2);
                SQLFreeConnect(o_hdbc2);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxnodcl");

        }

}

//================================================================================
//
// Function   : LoadOrderLineTable
//
//================================================================================

void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
        int     i,j;
  long  o_id;
        short   o_d_id;
        short   o_w_id;
  long  ol;
        long    ol_i_id;
  short ol_supply_w_id;
  short ol_quantity;
  double ol_amount;
  char  ol_dist_info[DIST_INFO_LEN+1];
        char            ol_delivery_d[OL_DELIVERY_D_LEN+1];
        RETCODE         rc;
        DBINT           rcint;

        // bind ORDER-LINE data
  rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

  rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 2);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

  rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

  rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

  rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

  rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0, OL_DELIVERY_D_LEN, NULL, 0,
SQLCHARACTER, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

  rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

  rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
```

```
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

  rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL, 0, 0, 10);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        for (i = 0; i < orders_per_district; i++)
  {
                o_id    = orders_buf[i].o_id;
                o_d_id  = orders_buf[i].o_d_id;
                o_w_id  = orders_buf[i].o_w_id;

                for (j=0; j < orders_buf[i].o_ol_cnt; j++)
                {
                        ol          = orders_buf[i].o_ol[j].ol;
                        ol_i_id     = orders_buf[i].o_ol[j].ol_i_id;
                        ol_supply_w_id = orders_buf[i].o_ol[j].ol_supply_w_id;
                        ol_quantity   = orders_buf[i].o_ol[j].ol_quantity;
                        ol_amount     = orders_buf[i].o_ol[j].ol_amount;
                        strcpy(ol_delivery_d,orders_buf[i].o_ol[j].ol_delivery_d);

                        strcpy(ol_dist_info,orders_buf[i].o_ol[j].ol_dist_info);

                        rc = bcp_sendrow(o_hdbc3);
                        if (rc != SUCCEED)
                                HandleErrorDBC(o_hdbc3);

                        order_line_rows_loaded++;
                        CheckForCommit(o_hdbc3, o_hstmt3, order_line_rows_loaded, "order_line",
&order_line_time_start->time_start);
                }

        }

        // rcint = bcp_batch(o_hdbc3);
        // if (rcint < 0)
        //      HandleErrorDBC(o_hdbc3);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc3);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc3);

                SQLFreeStmt(o_hstmt3, SQL_DROP);
                SQLDisconnect(o_hdbc3);
                SQLFreeConnect(o_hdbc3);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxodlcl");

        }

}
```

```
//==============================================================
//
// Function   : GetPermutation
//
//==============================================================

void GetPermutation(int perm[], int n)
{
        int i, r, t;

        for (i=1;i<=n;i++)
                perm[i] = i;

        for (i=1;i<=n;i++)
        {
                r = RandomNumber(i,n);
                t = perm[i];
                perm[i] = perm[r];
                perm[r] = t;
        }

}


//==============================================================
//
// Function   : CheckForCommit
//
//==============================================================

void CheckForCommit(HDBC hdbc,
                                        HSTMT hstmt,
                                int rows_loaded,
                                        char *table_name,
                                long *time_start)
{

  long      time_end, time_diff;
        // DBINT    rcint;

  if ( !(rows_loaded % aptr->batch) )
  {

                // rcint = bcp_batch(hdbc);
                // if (rcint < 0)
                //      HandleErrorDBC(hdbc);

                time_end = (TimeNow() / MILLI);
                time_diff = time_end - *time_start;

                printf("-> Loaded %ld rows into %s in %ld sec - Total = %d (%.2f rps)\n",
                        aptr->batch,
                        table_name,
                                time_diff,
```

```
                                        rows_loaded,
                                                (float) aptr->batch / (time_diff? time_diff: 1L));

                                *time_start = time_end;
    }

    return;
}


//==========================================================================
//
// Function   : OpenConnections
//
//==========================================================================

void OpenConnections()
{

        RETCODE         rc;

        char                            szDriverString[300];
        char                            szDriverStringOut[1024];
        SQLSMALLINT                     cbDriverStringOut;


        SQLAllocHandle(SQL_HANDLE_ENV,SQL_NULL_HANDLE, &henv );

        SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );

        SQLAllocHandle(SQL_HANDLE_DBC,henv , &i_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC,henv , &w_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC,henv , &c_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC,henv , &c_hdbc2);
        SQLAllocHandle(SQL_HANDLE_DBC,henv , &o_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC,henv , &o_hdbc2);
        SQLAllocHandle(SQL_HANDLE_DBC,henv , &o_hdbc3);

        SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );

        // Open connections to SQL Server

        // Connection 1

        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                                aptr->server,
                                                                aptr->user,
                                                                aptr->password,
                                                                aptr->database );

        rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = SQLDriverConnect ( i_hdbc1,
                                                                NULL,
                                                                (SQLCHAR*)&szDriverString[0],
                                                                SQL_NTS,
                                                                (SQLCHAR*)&szDriverStringOut[0],
                                                                sizeof(szDriverStringOut),
                                                                &cbDriverStringOut,
                                                                SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        // Connection 2

        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                                aptr->server,
                                                                aptr->user,
                                                                aptr->password,
                                                                aptr->database );

        rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = SQLDriverConnect ( w_hdbc1,
                                                                NULL,
                                                                (SQLCHAR*)&szDriverString[0]
,
                                                                SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],
                                                                sizeof(szDriverStringOut),
                                                                &cbDriverStringOut,
                                                                SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        // Connection 3

        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                                aptr->server,
                                                                aptr->user,
                                                                aptr->password,
                                                                aptr->database );

        rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

        rc = SQLDriverConnect ( c_hdbc1,
                                                                NULL,
                                                                (SQLCHAR*)&szDriverString[0]
,
```

```
                                                                SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
                                                                sizeof(szDriverStringOut),
                                                                &cbDriverStringOut,
                                                                SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

        // Connection 4

        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                                aptr->server,
                                                                aptr->user,
                                                                aptr->password,
                                                                aptr->database );

        rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        rc = SQLDriverConnect ( c_hdbc2,
                                                                NULL,
                                                                (SQLCHAR*)&szDriverString[0]
,
                                                                SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
                                                                sizeof(szDriverStringOut),
                                                                &cbDriverStringOut,
                                                                SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        // Connection 5

        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                                aptr->server,
                                                                aptr->user,
                                                                aptr->password,
                                                                aptr->database );

        rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        rc = SQLDriverConnect ( o_hdbc1,
                                                                NULL,
                                                                (SQLCHAR*)&szDriverString[0]
,
                                                                SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
                                                                sizeof(szDriverStringOut),
                                                                &cbDriverStringOut,
                                                                SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        // Connection 6

        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                                aptr->server,
                                                                aptr->user,
                                                                aptr->password,
                                                                aptr->database );

        rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        rc = SQLDriverConnect ( o_hdbc2,
                                                                NULL,
                                                                (SQLCHAR*)&szDriverString[0]
,
                                                                SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
                                                                sizeof(szDriverStringOut),
                                                                &cbDriverStringOut,
                                                                SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        // Connection 7

        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                                aptr->server,
                                                                aptr->user,
                                                                aptr->password,
                                                                aptr->database );

        rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        rc = SQLDriverConnect ( o_hdbc3,
                                                                NULL,
                                                                (SQLCHAR*)&szDriverString[0]
,
                                                                SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
                                                                sizeof(szDriverStringOut),
                                                                &cbDriverStringOut,
                                                                SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

}
//================================================================
```

```
//
// Function name: BuildIndex
//
//============================================================================

void BuildIndex(char      *index_script)
{
        char            cmd[256];

        printf("Starting index creation:  %s\n",index_script);

        sprintf(cmd, "isql -S%s -U%s -P%s -e -i%s\\%s.sql > logs\\%s.log",
                                                aptr->server,
                                                aptr->user,
                                                aptr->password,
                                                aptr->index_script_path,
                                                index_script,
                                                index_script);

        system(cmd);

        printf("Finished index creation:  %s\n",index_script);
}


void HandleErrorDBC (SQLHDBC  hdbc1)
{

        SQLCHAR                 SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER  NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char                    timebuf[128];
        char                    datebuf[128];
        FILE                    *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i, SqlState , &NativeError,
                                                Msg, sizeof(Msg) , &MsgLen )) != SQL_NO_DATA )
        {

                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

                fp1 = fopen("logs\\tpccldr.err","w");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open errorlog file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf, szLastError);
                        fclose(fp1);
                }

                i++;

        }
}

void HandleErrorSTMT (HSTMT  hstmt1)
{

        SQLCHAR                 SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER  NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char                    timebuf[128];
        char                    datebuf[128];
        FILE                    *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i, SqlState , &NativeError,
                                                Msg, sizeof(Msg) , &MsgLen )) != SQL_NO_DATA )
        {

                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

                fp1 = fopen("logs\\tpccldr.err","w");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open errorlog file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf, szLastError);
                        fclose(fp1);
                }

                i++;

        }
}


void FormatDate ( char* szTimeCOutput )
{

        struct tm when;
        time_t now;

        time( &now );
        when  = *localtime( &now );

        mktime( &when );

        // odbc datetime format
```

```
	strftime( szTimeCOutput , 30 , "%Y-%m-%d %H:%M:%S.000", &when );

	return;

}


//====================================================================
//
// Function   : CheckSQL
//
//====================================================================

void CheckSQL()
{

	RETCODE		rc;

	char			szDriverString[300];
	char			szDriverStringOut[1024];
	int				SQLBuildFlag;

	SQLSMALLINT		cbDriverStringOut;
	SQLCHAR			SQLVersion[19];
	SQLINTEGER		SQLVersionInd;



	SQLAllocHandle(SQL_HANDLE_ENV,SQL_NULL_HANDLE, &henv );

	SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );

	SQLAllocHandle(SQL_HANDLE_DBC,henv , &v_hdbc);

	SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );

	// Open connection to SQL Server

	sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s" ,
							aptr->server,
							aptr->user,
							aptr->password );


	if ( SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE, (SQLPOINTER)aptr->pack_size,
SQL_IS_UINTEGER ) != SQL_SUCCESS )
		HandleErrorDBC(v_hdbc);

	rc = SQLDriverConnect ( v_hdbc,
					NULL,
					(SQLCHAR*)&szDriverString[0],
					SQL_NTS,
					(SQLCHAR*)&szDriverStringOut[0],
					sizeof(szDriverStringOut),
					&cbDriverStringOut,
					SQL_DRIVER_NOPROMPT );

	if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
		HandleErrorDBC(v_hdbc);

	if ( SQLAllocHandle(SQL_HANDLE_STMT,v_hdbc , &v_hstmt) != SQL_SUCCESS )
		HandleErrorSTMT(v_hstmt);

	rc = SQLBindCol(v_hstmt, 4, SQL_C_CHAR, &SQLVersion, sizeof(SQLVersion), &SQLVersionInd);

	// issue SQL Server extended stored procedure (xp_msver) to determine installed version
	rc = SQLExecDirect(v_hstmt, "EXECUTE xp_msver ProductVersion", SQL_NTS);

	if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
		HandleErrorSTMT(v_hstmt);

	rc = SQLFetch(v_hstmt);

	if (rc != SQL_SUCCESS)
		HandleErrorDBC(v_hdbc);

	// Check build number to ensure 7.00.623 or higher

	SQLBuildFlag= 1;

	if ( SQLVersion[0] == 55 )
	{
		if ( SQLVersion[2] == 48 )
		{
			if ( SQLVersion[5] == 56 )
			{
				if ( (SQLVersion[6] >= 48) &  (SQLVersion[7] >= 53) )
				{
					SQLBuildFlag= 0;
					printf("You are using SQL Server version = %9s\n\n",
SQLVersion);
				}
				else
				{
					SQLBuildFlag= 1;
				}
			}
			else
			{
				if ( SQLVersion[5] >= 54 )
				{
					if ( (SQLVersion[6] >= 50) & (SQLVersion[7] >= 51) )
					{
						SQLBuildFlag= 0;
						printf("You are using SQL Server version =
%9s\n\n", SQLVersion);
					}
					else
					{
						SQLBuildFlag= 1;
					}
				}
				else
```

```
                                        {
                                                if ( SQLVersion[5] >= 55 )
                                                {
                                                        if ( (SQLVersion[6] >= 48) & (SQLVersion[7]
>= 48) )
                                                        {
                                                                SQLBuildFlag= 0;
                                                                printf("You are using SQL Server
version = %9s\n\n", SQLVersion);
                                                        }
                                                        else
                                                        {
                                                                SQLBuildFlag= 1;
                                                        }
                                                }
                                        }
                                }
                                else
                                {
                                        if ( SQLVersion[5] >= 49 )
                                        {
                                                if ( (SQLVersion[6] >= 52) & (SQLVersion[7] >= 48) )
                                                {
                                                        SQLBuildFlag= 0;
                                                        printf("You are using SQL Server version = %9s\n\n",
SQLVersion);
                                                }
                                                else
                                                {
                                                        SQLBuildFlag= 1;
                                                }
                                        }
                                        else
                                        {
                                                SQLBuildFlag= 1;
                                        }
                                }
                        }
                        else
                        {
                                SQLBuildFlag= 1;
                        }


                        if ( SQLBuildFlag== 1 )
                        {
                                printf("ERROR.  The SQL Server version you are using is not supported\n");
                                printf("for TPC-C benchmarking.  You currently have SQL Server version %9s\n",SQLVersion);
                                printf("installed.  Please upgrade to Microsoft SQL Server 7.00.623 or better.\n");
                                printf("and re-run the SETUP program.\n\n");
                                exit(1);
                        }

                        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                        SQLDisconnect(v_hdbc);
                        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                        return;

}

//=============================================================================
//
// Function   : CheckDataBase
//
//=============================================================================

void CheckDataBase()
{

        RETCODE         rc;

        char                            szDriverString[300];
        char                            szDriverStringOut[1024];
        char                            TablesBitMap[9] = {"000000000"};
        int                                     i, ExitFlag;

        SQLSMALLINT                     cbDriverStringOut;
        SQLCHAR                         TabName[10];
        SQLINTEGER                      TabNameInd, TabCount, TabCountInd;


        ExitFlag= 0;

        SQLAllocHandle(SQL_HANDLE_ENV,SQL_NULL_HANDLE, &henv );

        SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );

        SQLAllocHandle(SQL_HANDLE_DBC,henv , &v_hdbc);

        SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );

        // Open connection to SQL Server

        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                                        aptr->server,
                                                                        aptr->user,
                                                                        aptr->password,
                                                                        aptr->database );


        rc = SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE, (SQLPOINTER)aptr->pack_size,
SQL_IS_UINTEGER );
        if (rc != SQL_SUCCESS)
                HandleErrorDBC(v_hdbc);

        rc = SQLDriverConnect ( v_hdbc,
                                                                NULL,
                                                                        (SQLCHAR*)&szDriverString[0],
                                                                        SQL_NTS,
                                                                        (SQLCHAR*)&szDriverStringOut[0],
```

```
                                                sizeof(szDriverStringOut),
                                                &cbDriverStringOut,
                                                SQL_DRIVER_NOPROMPT );

        // if the rc is SQL_ERROR, the the TPCC database probably does not exist
        if (rc == SQL_ERROR)
        {
                printf("The database TPCC does not appear to exist!\n");
                printf("\nCheck LOGS\\ directory for database creation errors.\n");

                // cleanup database connections and handles
                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                SQLDisconnect(v_hdbc);
                SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                // since there is not a database, exit back to SETUP.CMD
                exit(1);
        }

        if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt) != SQL_SUCCESS )
                HandleErrorDBC(v_hdbc);

        if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0, &TabCountInd) != SQL_SUCCESS )
                HandleErrorSTMT(v_hstmt);

        // count the number of user tables from sysobjects
        rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects where xtype = \'U\'", SQL_NTS);
        if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
                HandleErrorSTMT(v_hstmt);

        if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
                HandleErrorSTMT(v_hstmt);

        // if the number of tables is less than 9, select all the user tables in TPCC
        if (TabCount != 9)
        {
                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);

                SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt);

                if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR, &TabName, sizeof(TabName), &TabNameInd) !=
SQL_SUCCESS )
                        HandleErrorSTMT(v_hstmt);

                // select the list of user tables into a result set
                rc = SQLExecDirect(v_hstmt, "select * from sysobjects where xtype = \'U\'", SQL_NTS);
                if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
                        HandleErrorSTMT(v_hstmt);

                // go through the result set and set the bitmap for each found table
                // set the bitmap to '1' if the table name is found

                while ((rc = SQLFetch(v_hstmt)) != SQL_NO_DATA)
                {
                        switch( TabName[0] )
                        {
                                case 'w':
                                        TablesBitMap[0] = '1';
                                        break;
                                case 'd':
                                        TablesBitMap[1] = '1';
                                        break;
                                case 'c':
                                        TablesBitMap[2] = '1';
                                        break;
                                case 'h':
                                        TablesBitMap[3] = '1';
                                        break;
                                case 'n':
                                        TablesBitMap[4] = '1';
                                        break;
                                case 'o':
                                        if (TabName[5] = 's')
                                                TablesBitMap[5] = '1';
                                        if (TabName[5] = '_')
                                                TablesBitMap[6] = '1';
                                        break;
                                case 'i':
                                        TablesBitMap[7] = '1';
                                        break;
                                case 's':
                                        TablesBitMap[8] = '1';
                                        break;
                        }
        }

        // a '0' ExitFlag means do NOT exit the loader early, a '1' means exit the loader early
        ExitFlag = 0;

        // interate through the bitmap to display which table(s) is actually missing
        for (i = 0; i <= 8; i++)
        {
                switch(i)
                {
                case 0:
                        if (TablesBitMap[i] == '0')
                        {
                                printf("The Warehouse table is missing or damaged.\n");
                                ExitFlag = 1;
                        }
                        break;
                case 1:
                        if (TablesBitMap[i] == '0')
                        {
                                printf("The District table is missing or damaged.\n");
                                ExitFlag = 1;
                        }
                        break;
                case 2:
                        if (TablesBitMap[i] == '0')
                        {
                                printf("The Customer table is missing or damaged.\n");
```

```
                        ExitFlag = 1;
                }
                break;
        case 3:
                if (TablesBitMap[i] == '0')
                {
                        printf("The History table is missing or damaged.\n");
                        ExitFlag = 1;
                }
                break;
        case 4:
                if (TablesBitMap[i] == '0')
                {
                        printf("The New_Order table is missing or damaged.\n");
                        ExitFlag = 1;
                }
                break;
        case 5:
                if (TablesBitMap[i] == '0')
                {
                        printf("The Orders table is missing or damaged.\n");
                        ExitFlag = 1;
                }
                break;
        case 6:
                if (TablesBitMap[i] == '0')
                {
                        printf("The Order_Line table is missing or damaged.\n");
                        ExitFlag = 1;
                }
                break;
        case 7:
                if (TablesBitMap[i] == '0')
                {
                        printf("The Item table is missing or damaged.\n");
                        ExitFlag = 1;
                }
                break;
        case 8:
                if (TablesBitMap[i] == '0')
                {
                        printf("The Stock table is missing or damaged.\n");
                        ExitFlag = 1;
                }
                break;
        }
}

// if one or more tables are missing, display message and exit the loader
if (ExitFlag = 1)
{
        printf("\nExiting TPC-C Loader!\n");
        printf("\nCheck LOGS\\ directory for database\n");
        printf("or table creation errors.\n");

        // cleanup database connections and handles
                        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                        SQLDisconnect(v_hdbc);
                        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                        exit(1);
                }
        }

        // cleanup database connections and handles
        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
        SQLDisconnect(v_hdbc);
        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

        return;
}
```

getargs.c

```
//      File:                   GETARGS.C
//                                      Microsoft TPC-C Kit Ver. 4.20
//                                      Copyright Microsoft, 1996, 1997, 1998, 1999
//      Purpose:        Source file for command line processing


// Includes
#include "tpcc.h"

//==============================================================
//
// Function name: GetArgsLoader
//
//==============================================================

void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{
   int                  i;
   char         *ptr;

#ifdef DEBUG
   printf("[%ld]DBG: Entering GetArgsLoader()\n", (int) GetCurrentThreadId());
#endif

   /* init args struct with some useful values */
   pargs->server                = SERVER;
   pargs->user                  = USER;
   pargs->password              = PASSWORD;
   pargs->database              = DATABASE;
   pargs->batch                 = BATCH;
   pargs->num_warehouses                = UNDEF;
        pargs->tables_all                       = TRUE;
        pargs->table_item                       = FALSE;
        pargs->table_warehouse          = FALSE;
        pargs->table_customer           = FALSE;
        pargs->table_orders                     = FALSE;
```

```
        pargs->loader_res_file          = LOADER_RES_FILE;
        pargs->pack_size                = DEFLDPACKSIZE;
        pargs->starting_warehouse       = DEF_STARTING_WAREHOUSE;
        pargs->build_index                      = BUILD_INDEX;
        pargs->index_order                      = INDEX_ORDER;
        pargs->index_script_path= INDEX_SCRIPT_PATH;
        pargs->scale_down                       = SCALE_DOWN;

/* check for zero command line args */
if ( argc == 1 )
                GetArgsLoaderUsage();

for (i = 1; i < argc; ++i)
{
                if (argv[i][0] != '-' && argv[i][0] != '/')
                {
                printf("\nUnrecognized command");
                GetArgsLoaderUsage();
                exit(1);
                }

                ptr = argv[i];

                switch (ptr[1])
                {
                case 'h':    /* Fall throught */
                case 'H':
                                GetArgsLoaderUsage();
                                break;

                case 'D':
                                pargs->database = ptr+2;
                                break;

                case 'P':
                                pargs->password = ptr+2;
                                break;

                case 'S':
                                pargs->server = ptr+2;
                                break;

                case 'U':
                                pargs->user = ptr+2;
                                break;

                case 'b':
                                pargs->batch = atol(ptr+2);
                                break;

                case 'W':
                                pargs->num_warehouses = atol(ptr+2);
                                break;

                case 's':
                                pargs->starting_warehouse = atol(ptr+2);
                                break;

                case 't':
                                {
                                        pargs->tables_all = FALSE;
                                        if (strcmp(ptr+2,"item") == 0)
                                                pargs->table_item = TRUE;
                                        else if (strcmp(ptr+2,"warehouse") == 0)
                                                pargs->table_warehouse = TRUE;
                                        else if (strcmp(ptr+2,"customer") == 0)
                                                pargs->table_customer = TRUE;
                                        else if (strcmp(ptr+2,"orders") == 0)
                                                pargs->table_orders = TRUE;
                                        else
                                        {
                                        printf("\nUnrecognized command");
                                        GetArgsLoaderUsage();
                                        exit(1);
                                        }

                                        break;
                                }

                        case 'f':
                                pargs->loader_res_file = ptr+2;
                                break;

                        case 'p':
                                pargs->pack_size = atol(ptr+2);
                                break;

                        case 'i':
                                pargs->build_index = atol(ptr+2);
                                break;

                        case 'o':
                                pargs->index_order = atol(ptr+2);
                                break;

                        case 'c':
                                pargs->scale_down = atol(ptr+2);
                                break;

                case 'd':
                                pargs->index_script_path = ptr+2;
                                break;

                default:
                                GetArgsLoaderUsage();
                                exit(-1);
                                break;
                }
}

/* check for required args */
if (pargs->num_warehouses == UNDEF )
```

```
    {
                printf("Number of Warehouses is required\n");
                exit(-2);
    }

    return;
}


//==============================================================================
//
// Function name: GetArgsLoaderUsage
//
//==============================================================================

void GetArgsLoaderUsage()
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int) GetCurrentThreadId());
#endif


    printf("TPCCLDR:\n\n");
                printf("Parameter                                   Default\n");
                printf("-----------------------------------------------------------------\n");
    printf("-W Number of Warehouses to Load                 Required \n");
    printf("-S Server                                   %s\n", SERVER);
    printf("-U Username                                  %s\n", USER);
    printf("-P Password                                 %s\n", PASSWORD);
    printf("-D Database                                 %s\n", DATABASE);
                printf("-b Batch Size                                %ld\n", (long) BATCH);
                printf("-p TDS packet size                           %ld\n", (long) DEFLDPACKSIZE);
                printf("-f Loader Results Output Filename                %s\n", LOADER_RES_FILE);
                printf("-s Starting Warehouse                        %ld\n", (long) DEF_STARTING_WAREHOUSE);
                printf("-i Build Option (data = 0, data and index = 1)       %ld\n", (long) BUILD_INDEX);
                printf("-o Cluster Index Build Order (before = 1, after = 0)  %ld\n", (long) INDEX_ORDER);
                printf("-c Build Scaled Database (normal = 0, tiny = 1)      %ld\n", (long) SCALE_DOWN);
                printf("-d Index Script Path                         %s\n", INDEX_SCRIPT_PATH);
                printf("-t Table to Load                            all tables \n");
    printf("   [item|warehouse|customer|orders]\n");
    printf("   Notes: \n");
    printf("    - the '-t' parameter may be included multiple times to \n");
    printf("      specify multiple tables to be loaded \n");
    printf("    - 'item' loads ITEM table \n");
    printf("    - 'warehouse' loads WAREHOUSE, DISTRICT, and STOCK tables \n");
    printf("    - 'customer' loads CUSTOMER and HISTORY tables \n");
    printf("    - 'orders' load NEW-ORDER, ORDERS, ORDER-LINE tables \n");

                printf("\nNote:  Command line switches are case sensitive.\n");

    exit(0);
}
```

time.c

```
//              File:                   TIME.C
//                                                      Microsoft TPC-C Kit Ver. 4.20
//                                                      Copyright Microsoft, 1996, 1997, 1998, 1999
//              Purpose:        Source file for time functions


// Includes
#include "tpcc.h"

// Globals
static long start_sec;


//==============================================================================
//
// Function name: TimeNow
//
//==============================================================================

long TimeNow()
{
  long                  time_now;
  struct        _timeb el_time;

#ifdef DEBUG
    printf("[%ld]DBG: Entering TimeNow()\n", (int) GetCurrentThreadId());
#endif

  _ftime(&el_time);

  time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

  return time_now;
}
```

strings.c

```
//              File:                   STRINGS.C
//                                                      Microsoft TPC-C Kit Ver. 4.20
//                                                      Copyright Microsoft, 1996, 1997, 1998, 1999
//              Purpose:        Source file for database loader string functions


// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//==============================================================================
//
// Function name: MakeAddress
```

```
//
//====================================================================

void MakeAddress(char *street_1,
                                                char *street_2,
                                                char *city,
                                                char *state,
                                                char *zip)
{

#ifdef DEBUG
   printf("[%ld]DBG: Entering MakeAddress()\n", (int) GetCurrentThreadId());
#endif

   MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
   MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
   MakeAlphaString (10, 20, ADDRESS_LEN, city);
   MakeAlphaString ( 2,  2, STATE_LEN, state);
   MakeZipNumberString( 9,  9, ZIP_LEN, zip);

#ifdef DEBUG
   printf("[%ld]DBG: MakeAddress: street_1: %s, street_2: %s, city: %s, state: %s, zip: %s\n",
                                        (int) GetCurrentThreadId(), street_1, street_2, city, state, zip);
#endif

   return;

}


//====================================================================
//
// Function name: LastName
//
//====================================================================

void LastName(int  num,
                                char *name)
{
   static char *n[] =
   {
                                "BAR" , "OUGHT", "ABLE" , "PRI"  , "PRES",
                                "ESE" , "ANTI" , "CALLY", "ATION", "EING"
   };

#ifdef DEBUG
   printf("[%ld]DBG: Entering LastName()\n", (int) GetCurrentThreadId());
#endif

   if ((num >= 0) && (num < 1000))
   {
                                strcpy(name, n[(num/100)%10]);
                                strcat(name, n[(num/10)%10]);
                                strcat(name, n[(num/1)%10]);

                                if (strlen(name) < LAST_NAME_LEN)
                                {
                                        PaddString(LAST_NAME_LEN,name);
                                }

   }
   else
   {
                                printf("\nError in LastName()... num <%ld> out of range (0,999)\n", num);
                                exit(-1);
   }


#ifdef DEBUG
   printf("[%ld]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
                                        (int) GetCurrentThreadId(), num, num/100, (num/10)%10, num%10);
                printf("[%ld]DBG: LastName: String = %s\n", (int) GetCurrentThreadId(), name);
#endif

   return;

}


//====================================================================
//
// Function name: MakeAlphaString
//
//====================================================================

//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2       The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a string of random alphanumeric
//(respectively, numeric) characters of a random length of minimum x, maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9.  The only other
//requirement is that the character set used "must be able to represent a minimum
//of 128 different characters".  We are using 8-bit chars, so this is a non issue.
//It is completely unreasonable to stuff non-printing chars into the text fields.
//-CLevine 08/13/96

int MakeAlphaString( int  x, int  y, int  z, char *str)
{
                int                             len;
                int                             i;
   char    cc = 'a';
                static          char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
                static          int             chArrayMax = 61;

#ifdef DEBUG
   printf("[%ld]DBG: Entering MakeAlphaString()\n", (int) GetCurrentThreadId());
#endif

                len= RandomNumber(x, y);
```

```
        for (i=0; i<len; i++)
        {
                cc = chArray[RandomNumber(0, chArrayMax)];
    str[i] = cc;
  }
        if ( len < z )
                memset(str+len, ' ', z - len);
        str[len] = 0;

  return len;
}


//====================================================================
//
// Function name: MakeOriginalAlphaString
//
//====================================================================

int MakeOriginalAlphaString(int x,
                                                        int  y,
                                                                int  z,
                                                        char *str,
                                                    int  percent)
{
  int           len;
  int           val;
  int           start;

#ifdef DEBUG
  printf("[%ld]DBG: Entering MakeOriginalAlphaString()\n",(int) GetCurrentThreadId());
#endif

  // verify prercentage is valid
  if ((percent < 0) || (percent > 100))
  {
                printf("MakeOrigianlAlphaString:Invalid percentage: %d\n", percent);
                exit(-1);
  }

  // verify string is at least 8 chars in length
  if ((x + y) <= 8)
  {
                printf("MakeOriginalAlphaString:string length must be >= 8\n");
                exit(-1);
  }

  // Make Alpha String
  len = MakeAlphaString(x,y, z, str);

  val = RandomNumber(1,100);
  if (val <= percent)
  {
                start = RandomNumber(0, len - 8);
                strncpy(str + start, "ORIGINAL", 8);
  }

#ifdef DEBUG
  printf("[%ld]DBG: MakeOriginalAlphaString:: %s\n",
                                        (int) GetCurrentThreadId(), str);
#endif

  return strlen(str);
}


//====================================================================
//
// Function name: MakeNumberString
//
//====================================================================
int MakeNumberString(int x, int  y, int  z, char *str)
{
        char tmp[16];

        //MakeNumberString is always called MakeZipNumberString(16, 16, 16, string)

        memset(str, '0', 16);
        itoa(RandomNumber(0, 99999999), tmp, 10);
        memcpy(str, tmp, strlen(tmp));

        itoa(RandomNumber(0, 99999999), tmp, 10);
        memcpy(str+8, tmp, strlen(tmp));

        str[16] = 0;

  return 16;
}


//====================================================================
//
// Function name: MakeZipNumberString
//
//====================================================================
int MakeZipNumberString(int x, int  y, int  z, char *str)
{
        char tmp[16];

        //MakeZipNumberString is always called MakeZipNumberString(9, 9, 9, string)

        strcpy(str, "000011111");

        itoa(RandomNumber(0, 9999), tmp, 10);
        memcpy(str, tmp, strlen(tmp));

  return 9;
}


//====================================================================
```

```
//
// Function name: InitString
//
//============================================================================
void InitString(char *str, int len)
{
#ifdef DEBUG
   printf("[%ld]DBG: Entering InitString()\n", (int) GetCurrentThreadId());
#endif

                memset(str, ' ', len);
   str[len] = 0;
}


//============================================================================
// Function name: InitAddress
//
// Description:
//
//============================================================================

void InitAddress(char *street_1, char *street_2, char *city, char *state, char *zip)
{
                memset(street_1, ' ', ADDRESS_LEN+1);
                memset(street_2, ' ', ADDRESS_LEN+1);
                memset(city, ' ', ADDRESS_LEN+1);

   street_1[ADDRESS_LEN+1] = 0;
   street_2[ADDRESS_LEN+1] = 0;
   city[ADDRESS_LEN+1]     = 0;

                memset(state, ' ', STATE_LEN+1);
   state[STATE_LEN+1] = 0;

                memset(zip, ' ', ZIP_LEN+1);
   zip[ZIP_LEN+1] = 0;
}


//============================================================================
//
// Function name: PaddString
//
//============================================================================

void PaddString(int max, char *name)
{
                int                             len;

                len = strlen(name);
                if ( len < max )
                                memset(name+len, ' ', max - len);
                name[max] = 0;

                return;
}
```

random.c

```
//              File:                           RANDOM.C
//                                                      Microsoft TPC-C Kit Ver. 4.20
//                                                      Copyright Microsoft, 1996, 1997, 1998, 1999
//              Purpose:        Random number generation routines for database loader


// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A           16807
#define M       2147483647
#define Q           127773      /* M div A */
#define R            2836       /* M mod A */
#define Thread              __declspec(thread)

// Globals
long            Thread Seed = 0;        /* thread local seed */

/*****************************************************************************
*                                                       *
* random -                                                      *
*       Implements a GOOD pseudo random number generator.  This generator      *
*       will/should? run the complete period before repeating.                  *
*                                                       *
* Copied from:                                                  *
*       Random Numbers Generators: Good Ones Are Hard to Find.                  *
*       Communications of the ACM - October 1988 Volume 31 Number 10            *
*                                                       *
* Machine Dependencies:                                         *
*       long must be 2 ^ 31 - 1 or greater.                             *
*                                                       *
*****************************************************************************/

/*****************************************************************************
* seed - load the Seed value used in irand and drand.  Should be used before  *
*       first call to irand or drand.                                   *
*****************************************************************************/

void seed(long val)
{

#ifdef DEBUG
   printf("[%ld]DBG: Entering seed()...\n", (int) GetCurrentThreadId());
                printf("Old Seed %ld New Seed %ld\n",Seed, val);
#endif

   if ( val < 0 )
                        val = abs(val);
}
```

```
    Seed = val;

}


/**************************************************************************
*                                                                         *
* irand - returns a 32 bit integer pseudo random number with a period of  *
*       1 to 2 ^ 32 - 1.                                                  *
*                                                                         *
* parameters:                                                             *
*       none.                                                             *
*                                                                         *
* returns:                                                                *
*       32 bit integer - defined as long ( see above ).                   *
*                                                                         *
* side effects:                                                           *
*       seed get recomputed.                                              *
**************************************************************************/

long irand()
{
    register long    s;     /* copy of seed */
    register long    test;  /* test flag */
    register long    hi;    /* tmp value for speed */
    register long    lo;    /* tmp value for speed */

#ifdef DEBUG
    printf("[%ld]DBG: Entering irand()...\n", (int) GetCurrentThreadId());
#endif

    s = Seed;
    hi = s / Q;
    lo = s % Q;

    test = A * lo - R * hi;
    if ( test > 0 )
                        Seed = test;
    else
                        Seed = test + M;

    return( Seed );
}


/**************************************************************************
*                                                                         *
* drand - returns a double pseudo random number between 0.0 and 1.0.      *
*       See irand.                                                        *
**************************************************************************/
double drand()
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering drand()...\n", (int) GetCurrentThreadId());
#endif

    return( (double)irand() / 2147483647.0);
}


//=========================================================================
// Function   : RandomNumber
//
// Description:
//=========================================================================
long RandomNumber(long lower, long upper)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering RandomNumber()...\n", (int) GetCurrentThreadId());
#endif

                if ( upper == lower )       /* pgd 08-13-96 perf enhancement */
                        return lower;

                upper++;

    if ( upper <= lower )
                        rand_num = upper;
    else
                        rand_num = lower + irand() % (upper - lower); /* pgd 08-13-96 perf enhancement */


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                                    (int) GetCurrentThreadId(), lower, upper, rand_num);
#endif


    return rand_num;
}


#if 0

//Orginal code pgd 08/13/96

long RandomNumber(long lower,
                                                    long upper)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering RandomNumber()...\n", (int) GetCurrentThreadId());
#endif

                upper++;

    if ((upper <= lower))
```

```
                        rand_num = upper;
    else
                        rand_num = lower + irand() % ((upper > lower) ? upper - lower : upper);


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                                (int) GetCurrentThreadId(), lower, upper, rand_num);
#endif


    return rand_num;
}
#endif

//============================================================================
// Function   : NURand
//
// Description:
//============================================================================
long NURand(int iConst,
                    long x,
                    long y,
                    long C)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering NURand()...\n", (int) GetCurrentThreadId());
#endif

    rand_num = (((RandomNumber(0,iConst) | RandomNumber(x,y)) + C) % (y-x+1))+x;

#ifdef DEBUG
    printf("[%ld]DBG: NURand: num = %d\n", (int) GetCurrentThreadId(), rand_num);
#endif

    return rand_num;
}
```

Appendix C: Tunable Parameters

Microsoft Windows NT 4.0 EE Configuration Parameters

The following services were disabled in the Windows NT Control Panel/Services on the Server:

- Computer Browser
- License Logging Service
- Messenger
- Plug and Play
- RPC Locator Service
- Spooler
- Server
- TCP/IP NetBIOS Helper

The following changes were made to the server configuration:

Set Control Panel -> System -> Performance -> Forground Application Boost to Minimum

Set Control Panel -> Neteork -> Services -> Server -> Maximize Throughput For Network Applications

Windows NT Server 4.0 Service Pack 6 was applied to the server.

Microsoft SQL Server 7.0 Startup Parameters

c:\sqlservr -c -x -T3502 -g38

Where:

- -c Start SQL Server independently of the Service Control Manager
- -x Disables the keeping of CPU time and cache hit ratio statistics
- -T3502 Writes a message to the SQL Server Errorlog showing the beginning and ending time of each checkpoint
- -g38 Specifies the amount of memory that is set aside for allocations not from the buffer pool

SQL Server Stack Size

The default stack size for Microsoft SQL Server 7.0 Enterprise Edition (7.00.805) was changed using the EDITBIN utility. The EDITBIN utility ships with Microsoft Visual C++ V6.0. The command used to change the stack size is:

editbin / Stack:131072

This command is fully documented as an article in the Microsoft Knowledge Base on the Microsoft Web site at www.microsoft.com/support.

Boot.ini

The /3gb switch was added to the boot.ini file to cause NT Server to allow 3GB of user and 1GB of kernel virtual address space, rather than the usual 2GB of virtual address space for each.

Microsoft SQL Server 7.0 Configuration Parameters

```
1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11>
--  File:     VERSION.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Returns SQL Server version string



print " "
select convert(char(30), getdate(),9)
print " "

 ------------------------------
 Feb 20 2000  1:49:11:900PM

(1 row affected)

1> 2> 3>
select @@version




 ------------------------------------------------------------------------------
                 ------------------------------------------------------------------------------
                 ------------------------------------------------------------------------------
                 ----------------------------------
 Microsoft SQL Server  7.00 - 7.00.805 (Intel X86)
                 Jun 11 1999 11:48:12
                 Cop
                 yright (c) 1988-1998 Microsoft Corporation
                 Enterprise Edition on Windo
                 ws NT 4.0 (Build 1381: Service Pack 6)



(1 row affected)
1> 2>
1> 2> 3> 4> 5> 6> 7> 8> 9> 10>
--  File:     CONFIG.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.20
--            Copyright Microsoft, 1999
--  Purpose:  Collects SQL Server configuration parameters
```

```
print " "
select convert(char(30), getdate(),9)
print " "

 ------------------------------
 Feb 20 2000  1:49:12:727PM

(1 row affected)

1> 2> 3> DBCC execution completed. If DBCC printed error messages, contact your system administrator.
Configuration option changed. Run the RECONFIGURE statement to install.

sp_configure "show advanced",1
1> 2> reconfigure with override
1> 2> sp_configure
 name                                 minimum     maximum     config_value run_value
 ------------------------------------ ----------- ----------- ------------ -----------
 affinity mask                                  0  2147483647          255         255
 allow updates                                  0           1            0           0
 cost threshold for parallelism                 0       32767            5           5
 cursor threshold                              -1  2147483647           -1          -1
 default language                               0        9999            0           0
 default sortorder id                           0         255           50          50
 extended memory size (MB)                      0  2147483647            0           0
 fill factor (%)                                0         100            0           0
 index create memory (KB)                     704     1600000            0           0
 language in cache                              3         100            3           3
 language neutral full-text                     0           1            0           0
 lightweight pooling                            0           1            1           1
 locks                                       5000  2147483647            0           0
 max async IO                                   1         255          255         255
 max degree of parallelism                      0          32            1           1
 max server memory (MB)                         4  2147483647         3025        3025
 max text repl size (B)                         0  2147483647        65536       65536
 max worker threads                            10        1024          450         450
 media retention                                0         365            0           0
 min memory per query (KB)                    512  2147483647          512         512
 min server memory (MB)                         0  2147483647         3025        3025
 nested triggers                                0           1            1           1
 network packet size (B)                      512       65535         4096        4096
 open objects                                   0  2147483647            0           0
 priority boost                                 0           1            1           1
 query governor cost limit                      0  2147483647            0           0
 query wait (s)                                -1  2147483647           -1          -1
 recovery interval (min)                        0       32767        32767       32767
 remote access                                  0           1            0           0
 remote login timeout (s)                       0  2147483647            5           5
 remote proc trans                              0           1            0           0
 remote query timeout (s)                       0  2147483647            0           0
 resource timeout (s)                           5  2147483647           10          10
 scan for startup procs                         0           1            0           0
 set working set size                           0           1            1           1
 show advanced options                          0           1            1           1
 spin counter                                   1  2147483647        10000       10000
 time slice (ms)                               50        1000          100         100
 two digit year cutoff                       1753        9999         2049        2049
 Unicode comparison style                       0  2147483647            0           0
 Unicode locale id                              0  2147483647        33280       33280
 user connections                               0       32767            0           0
 user options                                   0        4095            0           0

1>
```

Microsoft Windows NT Server Version 4.0 Configuration Parameters

```
Microsoft Diagnostics Report For \\IBMSERV3
------------------------------------------------------------------------------

OS Version Report
------------------------------------------------------------------------------
Microsoft (R) Windows NT (TM) Server
Version 4.0 (Build 1381: Service Pack 6) x86 Multiprocessor Free
Registered Owner: TPC-C, IBM
Product Number: 70234-012-0123456-18037
------------------------------------------------------------------------------

System Report
------------------------------------------------------------------------------
System: AT/AT COMPATIBLE
Hardware Abstraction Layer: MPS 1.4 - APIC platform
BIOS Date: 02/18/00
BIOS Version: IBM BIOS Ver 2.0


Processor list:
  0:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
  1:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
  2:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
  3:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
  4:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
  5:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
  6:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
  7:  x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
------------------------------------------------------------------------------

Video Display Report
------------------------------------------------------------------------------
BIOS Date: 09/02/99
BIOS Version: S3 86C366 Video BIOS. Version 2.0B.05-C5.03.04a 83MHz
              S3 86C366 Video BIOS. Version 2.0B.05-C5.03.04a 83MHz
              S3 86C366 Video BIOS. Version 2.0B.05-C5.03.04a 83MHz

Adapter:
  Setting: 800 x 600 x 16
           Hardware Default Refresh
  Type: vga compatible display adapter
  String: <unavailable>
```

```
  Memory:
  Chip Type: <unavailable>
  DAC Type: <unavailable>
Driver:
  Vendor: Microsoft Corporation
  File(s): vga.sys, vga.dll
  Version: 4.00, 4.0.0


Drives Report
-----------------------------------------------------------------------
C:\  (Local - NTFS)  Total: 4,192,933 KB, Free: 2,787,831 KB
  Serial Number: 40A2 - 7CFB
  Bytes per cluster: 512
  Sectors per cluster: 1
  Filename length: 255
D:\  (CDROM - CDFS) NETFAPP1 Total: 512,200 KB, Free: 0 KB
  Serial Number: B69D - 2A4
  Bytes per cluster: 2048
  Sectors per cluster: 1
  Filename length: 221
Z:\  (Local - NTFS) backup1 Total: 325,816,316 KB, Free: 73,520,992 KB
  Serial Number: 40B4 - 69E
  Bytes per cluster: 512
  Sectors per cluster: 8
  Filename length: 255

Memory Report
-----------------------------------------------------------------------
Handles: 1,561
Threads: 110
Processes: 17

Physical Memory (K)
  Total: 3,931,524
  Available: 2,471,656
  File Cache: 15,000

Kernel Memory (K)
  Total: 13,028
  Paged: 9,528
  Nonpaged: 3,500

Commit Charge (K)
  Total: 1,326,560
  Limit: 4,050,788
  Peak: 3,093,108

Pagefile Space (K)
  Total: 273,408
  Total in use: 3,124
  Peak: 3,648

  C:\pagefile.sys
    Total: 273,408
    Total in use: 3,124
    Peak: 3,648


Services Report
-----------------------------------------------------------------------
Alerter                         Running   (Automatic)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation
Computer Browser                Stopped   (Manual)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation
    LanmanServer
    LmHosts
ClipBook Server                 Stopped   (Manual)
  C:\WINNT\system32\clipsrv.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
  Service Dependencies:
    NetDDE
DHCP Client (TDI)               Stopped   (Disabled)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    Tcpip
    Afd
    NetBT
Disk Array Monitor              Stopped   (Manual)
  C:\Program Files\SYMsm\arraymon.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
EventLog (Event log)            Running   (Automatic)
  C:\WINNT\system32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
Gopher Publishing Service       Stopped   (Manual)
  C:\WINNT\System32\inetsrv\inetinfo.exe
  Service Account Name: LocalSystem
  Error Severity: Ignore
  Service Flags: Shared Process
  Service Dependencies:
    RPCSS
    NTLMSSP
```

```
Server                          Stopped   (Manual)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Group Dependencies:
    TDI
Workstation (NetworkProvider)   Running   (Automatic)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Group Dependencies:
    TDI
License Logging Service         Stopped   (Manual)
  C:\WINNT\System32\llssrv.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
TCP/IP NetBIOS Helper           Stopped   (Manual)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Group Dependencies:
    NetworkProvider
Messenger                       Stopped   (Manual)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation
    NetBios
MSDTC (MS Transactions)         Stopped   (Manual)
  C:\WINNT\System32\msdtc.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
  Service Dependencies:
    RPCSS
    NTLMSSP
FTP Publishing Service          Stopped   (Manual)
  C:\WINNT\System32\inetsrv\inetinfo.exe
  Service Account Name: LocalSystem
  Error Severity: Ignore
  Service Flags: Shared Process
  Service Dependencies:
    RPCSS
    NTLMSSP
MSSQLServer                     Stopped   (Manual)
  C:\MSSQL7\binn\sqlservr.exe
  Service Account Name: .\Administrator
  Error Severity: Normal
  Service Flags: Own Process
Network DDE (NetDDEGroup)       Stopped   (Manual)
  C:\WINNT\system32\netdde.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    NetDDEDSDM
Network DDE DSDM                Stopped   (Manual)
  C:\WINNT\system32\netdde.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
Net Logon (RemoteValidation)    Stopped   (Manual)
  C:\WINNT\System32\lsass.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation
    LmHosts
NT LM Security Support Provider Running   (Manual)
  C:\WINNT\System32\SERVICES.EXE
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
Plug and Play (PlugPlay)        Stopped   (Manual)
  C:\WINNT\system32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
Protected Storage               Running   (Automatic)
  c:\winnt\system32\pstores.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process, Interactive
  Service Dependencies:
    RpcSs
Directory Replicator            Stopped   (Manual)
  C:\WINNT\System32\lmrepl.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
  Service Dependencies:
    LanmanWorkstation
    LanmanServer
Remote Procedure Call (RPC) Locator  Stopped   (Manual)
  C:\WINNT\System32\LOCATOR.EXE
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
  Service Dependencies:
    LanmanWorkstation
    Rdr
Remote Procedure Call (RPC) Service  Running   (Automatic)
  C:\WINNT\system32\RpcSs.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
```

```
   Service Flags: Own Process
Schedule                          Stopped   (Manual)
   C:\WINNT\System32\AtSvc.Exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
Spooler (SpoolerGroup)            Stopped   (Manual)
   C:\WINNT\system32\spoolss.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process, Interactive
SQLServerAgent                    Stopped   (Manual)
   C:\MSSQL7\binn\sqlagent.exe
   Service Account Name: .\Administrator
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      MSSQLServer
Telephony Service                 Stopped   (Manual)
   C:\WINNT\system32\tapisrv.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
UPS                               Stopped   (Manual)
   C:\WINNT\System32\ups.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
World Wide Web Publishing Service Stopped   (Manual)
   C:\WINNT\System32\inetsrv\inetinfo.exe
   Service Account Name: LocalSystem
   Error Severity: Ignore
   Service Flags: Shared Process
   Service Dependencies:
      RPCSS
      NTLMSSP



Drivers Report
--------------------------------------------------------------------------------
Abiosdsk (Primary disk)           Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
AFD Networking Support Environment(TDI)     Running   (Automatic)
   C:\WINNT\System32\drivers\afd.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Aha154x (SCSI miniport)           Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Aha174x (SCSI miniport)           Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
aic78u2 (SCSI miniport)           Running   (Boot)
   C:\WINNT\system32\drivers\aic78u2.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
aic78xx (SCSI miniport)           Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Always (SCSI miniport)            Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
ami0nt (SCSI miniport)            Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
amsint (SCSI miniport)            Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Arrow (SCSI miniport)             Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
atapi (SCSI miniport)             Running   (Boot)
   C:\WINNT\System32\DRIVERS\atapi.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Atdisk (Primary disk)             Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
ati (Video)                       Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Beep (Base)                       Running   (System)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
BusLogic (SCSI miniport)          Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Busmouse (Pointer Port)           Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Cdaudio (Filter)                  Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Cdfs (File system)                Running   (Disabled)
   Error Severity: Normal
   Service Flags: File System Driver, Shared Process
   Group Dependencies:
      SCSI CDROM Class
Cdrom (SCSI CDROM Class)          Running   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Changer (Filter)                  Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
cirrus (Video)                    Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Cpqarray (SCSI miniport)          Stopped   (Disabled)
```

Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
cpqfws2e (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
dac960nt (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
dce376nt (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Delldsa (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Dell_DGX (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Disk (SCSI Class) Running (Boot)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Group Dependencies:
SCSI miniport
Diskperf (Filter) Running (Boot)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
DptScsi (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
dtc329x (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
et4000 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Fastfat (Boot file system) Running (Disabled)
Error Severity: Normal
Service Flags: File System Driver, Shared Process
Fd16_700 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Fd7000ex (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Fd8xx (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
flashpnt (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Floppy (Primary disk) Running (System)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Ftdisk (Filter) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
i8042 Keyboard and PS/2 Mouse Port Driver (Keyboard Port) Running (System)
System32\DRIVERS\i8042prt.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
ibmcomw Running (Automatic)
C:\WINNT\System32\Drivers\ibmcomw.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
IBM(R) Netfinity NDIS Driver (NDIS) Running (Automatic)
C:\WINNT\System32\drivers\IBMFENT.SYS
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
ibmspw Running (Automatic)
C:\WINNT\System32\Drivers\ibmspw.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Inport (Pointer Port) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
intlfxsr (Base) Running (Boot)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Jazzg300 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Jazzg364 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Jzvxl484 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Keyboard Class Driver (Keyboard Class) Running (System)
System32\DRIVERS\kbdclass.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
KSecDD (Base) Running (System)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
mga (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
mga_mil (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
mitsumi (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
mkecr5xx (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Modem (Extended base) Stopped (Manual)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Mouse Class Driver (Pointer Class) Running (System)
System32\DRIVERS\mouclass.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process

```
mraid (Primary disk)                    Running   (Boot)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Msfs (File system)                      Running   (System)
  Error Severity: Normal
  Service Flags: File System Driver, Shared Process
Mup (Network)                           Running   (Manual)
  C:\WINNT\System32\drivers\mup.sys
  Error Severity: Normal
  Service Flags: File System Driver, Shared Process
Ncr53c9x (SCSI miniport)                Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
ncr77c22 (Video)                        Stopped   (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Ncrc700 (SCSI miniport)                 Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Ncrc710 (SCSI miniport)                 Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Microsoft NDIS System Driver (NDIS)     Running   (System)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
NetBIOS Interface (NetBIOSGroup)        Stopped   (Manual)
  C:\WINNT\System32\drivers\netbios.sys
  Error Severity: Normal
  Service Flags: File System Driver, Shared Process
  Group Dependencies:
    TDI
WINS Client(TCP/IP) (PNP_TDI)           Running   (Automatic)
  C:\WINNT\System32\drivers\netbt.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
  Service Dependencies:
    Tcpip
NetDetect                               Stopped   (Manual)
  C:\WINNT\system32\drivers\netdtect.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Npfs (File system)                      Running   (System)
  Error Severity: Normal
  Service Flags: File System Driver, Shared Process
Ntfs (File system)                      Running   (Disabled)
  Error Severity: Normal
  Service Flags: File System Driver, Shared Process
Null (Base)                             Running   (System)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Oliscsi (SCSI miniport)                 Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Parallel (Extended base)                Stopped   (Automatic)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
  Service Dependencies:
    Parport
  Group Dependencies:
    Parallel arbitrator
Parport (Parallel arbitrator)           Stopped   (Automatic)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
ParVdm (Extended base)                  Stopped   (Automatic)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
  Service Dependencies:
    Parport
  Group Dependencies:
    Parallel arbitrator
PCIDump (PCI Configuration)             Stopped   (System)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Pcmcia (System Bus Extender)            Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
PnP ISA Enabler Driver (Base)           Stopped   (System)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
psidisp (Video)                         Stopped   (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Ql10wnt (SCSI miniport)                 Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
ql2100 (SCSI Miniport)                  Running   (Boot)
  C:\WINNT\System32\DRIVERS\ql2100.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
qv (Video)                              Stopped   (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Rdr (Network)                           Running   (Manual)
  C:\WINNT\System32\drivers\rdr.sys
  Error Severity: Normal
  Service Flags: File System Driver, Shared Process
s3 (Video)                              Stopped   (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Scsiprnt (Extended base)                Stopped   (Automatic)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
  Group Dependencies:
    SCSI miniport
Scsiscan (SCSI Class)                   Running   (System)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
  Group Dependencies:
    SCSI miniport
Serial (Extended base)                  Running   (Automatic)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
```

Sermouse (Pointer Port) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Sfloppy (Primary disk) Stopped (System)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Group Dependencies:
SCSI miniport
Simbad (Filter) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
slcd32 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Sparrow (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Spock (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Srv (Network) Stopped (Manual)
C:\WINNT\System32\drivers\srv.sys
Error Severity: Normal
Service Flags: File System Driver, Shared Process
symarray (SCSI Class) Running (Boot)
Error Severity: Severe
Service Flags: Kernel Driver, Shared Process
Group Dependencies:
SCSI miniport

A?

Start
symc810 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
T128 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
T13B (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
TCP/IP Service (PNP_TDI) Running (Automatic)
C:\WINNT\System32\drivers\tcpip.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
tga (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
tmv1 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Ultra124 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Ultra14f (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Ultra24f (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
update (Base) Stopped (System)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
v7vram (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
VgaSave (Video Save) Running (System)
C:\WINNT\System32\drivers\vga.sys
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
VgaStart (Video Init) Stopped (System)
C:\WINNT\System32\drivers\vga.sys
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Wd33c93 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
wd90c24a (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
wdvga (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
weitekp9 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Xga (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process

IRQ and Port Report

Devices	Vector	Level	Affinity
MPS 1.4 - APIC platform	8	8	0x000000ff
MPS 1.4 - APIC platform	0	0	0x000000ff
MPS 1.4 - APIC platform	1	1	0x000000ff
MPS 1.4 - APIC platform	2	2	0x000000ff
MPS 1.4 - APIC platform	3	3	0x000000ff
MPS 1.4 - APIC platform	4	4	0x000000ff
MPS 1.4 - APIC platform	5	5	0x000000ff
MPS 1.4 - APIC platform	6	6	0x000000ff
MPS 1.4 - APIC platform	7	7	0x000000ff
MPS 1.4 - APIC platform	8	8	0x000000ff
MPS 1.4 - APIC platform	9	9	0x000000ff
MPS 1.4 - APIC platform	10	10	0x000000ff

```
MPS 1.4 - APIC platform        11   11 0x000000ff
MPS 1.4 - APIC platform        12   12 0x000000ff
MPS 1.4 - APIC platform        13   13 0x000000ff
MPS 1.4 - APIC platform        14   14 0x000000ff
MPS 1.4 - APIC platform        15   15 0x000000ff
MPS 1.4 - APIC platform        16   16 0x000000ff
MPS 1.4 - APIC platform        17   17 0x000000ff
MPS 1.4 - APIC platform        18   18 0x000000ff
MPS 1.4 - APIC platform        19   19 0x000000ff
MPS 1.4 - APIC platform        20   20 0x000000ff
MPS 1.4 - APIC platform        21   21 0x000000ff
MPS 1.4 - APIC platform        22   22 0x000000ff
MPS 1.4 - APIC platform        23   23 0x000000ff
MPS 1.4 - APIC platform        24   24 0x000000ff
MPS 1.4 - APIC platform        25   25 0x000000ff
MPS 1.4 - APIC platform        26   26 0x000000ff
MPS 1.4 - APIC platform        27   27 0x000000ff
MPS 1.4 - APIC platform        28   28 0x000000ff
MPS 1.4 - APIC platform        29   29 0x000000ff
MPS 1.4 - APIC platform        30   30 0x000000ff
MPS 1.4 - APIC platform        31   31 0x000000ff
MPS 1.4 - APIC platform        32   32 0x000000ff
MPS 1.4 - APIC platform        33   33 0x000000ff
MPS 1.4 - APIC platform        34   34 0x000000ff
MPS 1.4 - APIC platform        35   35 0x000000ff
MPS 1.4 - APIC platform        36   36 0x000000ff
MPS 1.4 - APIC platform        37   37 0x000000ff
MPS 1.4 - APIC platform        38   38 0x000000ff
MPS 1.4 - APIC platform        39   39 0x000000ff
MPS 1.4 - APIC platform        40   40 0x000000ff
MPS 1.4 - APIC platform        41   41 0x000000ff
MPS 1.4 - APIC platform        42   42 0x000000ff
MPS 1.4 - APIC platform        43   43 0x000000ff
MPS 1.4 - APIC platform        44   44 0x000000ff
MPS 1.4 - APIC platform        45   45 0x000000ff
MPS 1.4 - APIC platform        46   46 0x000000ff
MPS 1.4 - APIC platform        47   47 0x000000ff
MPS 1.4 - APIC platform        61   61 0x000000ff
MPS 1.4 - APIC platform        65   65 0x000000ff
MPS 1.4 - APIC platform        80   80 0x000000ff
MPS 1.4 - APIC platform       193  193 0x000000ff
MPS 1.4 - APIC platform       225  225 0x000000ff
MPS 1.4 - APIC platform       253  253 0x000000ff
MPS 1.4 - APIC platform       254  254 0x000000ff
MPS 1.4 - APIC platform       255  255 0x000000ff
i8042prt                        1    1  0xffffffff
i8042prt                       12   12  0xffffffff
ibmcomw                         8    8 0x00000000
Serial                          4    4 0x00000000
Floppy                          6    6 0x00000000
IBMFE                          12   12 0x00000000
aic78u2                        56   56 0x00000000
aic78u2                        56   56 0x00000000
atapi                           0   14 0x00000000
ql2100                         40   40 0x00000000
ibmspw                          8    8 0x00000000
```

Devices	Physical Address	Length
MPS 1.4 - APIC platform	0x00000000	0x0000000010
MPS 1.4 - APIC platform	0x00000020	0x0000000002
MPS 1.4 - APIC platform	0x00000040	0x0000000004
MPS 1.4 - APIC platform	0x00000048	0x0000000004
MPS 1.4 - APIC platform	0x00000061	0x0000000001
MPS 1.4 - APIC platform	0x00000070	0x0000000002
MPS 1.4 - APIC platform	0x00000080	0x0000000010
MPS 1.4 - APIC platform	0x00000092	0x0000000001
MPS 1.4 - APIC platform	0x000000a0	0x0000000002
MPS 1.4 - APIC platform	0x000000c0	0x0000000010
MPS 1.4 - APIC platform	0x000000f0	0x0000000010
i8042prt	0x00000060	0x0000000001
i8042prt	0x00000064	0x0000000001
ibmcomw	0x00002000	0x0000000100
ibmcomw	0x00002180	0x0000000010
Serial	0x000003f8	0x0000000007
Floppy	0x000003f0	0x0000000006
Floppy	0x000003f7	0x0000000001
IBMFE	0x0000a000	0x000000001e
aic78u2	0x00002300	0x0000000100
aic78u2	0x00002400	0x0000000100
atapi	0x000001f0	0x0000000008
atapi	0x000003f6	0x0000000001
ql2100	0x00002200	0x0000000100
ibmspw	0x00002000	0x0000000100
VgaSave	0x000003b0	0x000000000c
VgaSave	0x000003c0	0x0000000020
VgaSave	0x000001ce	0x0000000002

DMA and Memory Report

Devices	Channel	Port
Floppy	2	0

Devices	Physical Address	Length
MPS 1.4 - APIC platform	0xfec00000	0x00000400
MPS 1.4 - APIC platform	0xfee00000	0x00000400
IBMFE	0xfaeff000	0x0000001e
aic78u2	0xf7dfd000	0x00001000
aic78u2	0xf7dfc000	0x00001000
ql2100	0xf7dfe000	0x00001000
ibmspw	0xf7de0000	0x00010000
VgaSave	0x000a0000	0x00020000

Environment Report

System Environment Variables
  ComSpec=C:\WINNT\system32\cmd.exe
  Os2LibPath=C:\WINNT\system32\os2\dll;
  Path=C:\WINNT\system32;C:\WINNT;C:\MSSQL7\BINN;
  windir=C:\WINNT
  OS=Windows_NT
  PROCESSOR_ARCHITECTURE=x86
  PROCESSOR_LEVEL=6
  PROCESSOR_IDENTIFIER=x86 Family 6 Model 7 Stepping 3, GenuineIntel
  PROCESSOR_REVISION=0703
  NUMBER_OF_PROCESSORS=8
  ND_HOME=C:\Program Files\SYMsm
  ND_PATH=C:\Program Files\SYMsm

Environment Variables for Current User
  TEMP=C:\TEMP
  TMP=C:\TEMP

Network Report

Your Access Level: Admin & Local
Workgroup or Domain: WORKGROUP
Network Version: 4.0
LanRoot: WORKGROUP
Logged On Users: 1
Current User (1): Administrator
  Logon Domain: IBMSERV3
  Logon Server: IBMSERV3

Transport: NetBT_IBMFE1, 00-04-AC-36-44-FE, VC's: 0, Wan: Wan

Character Wait: 3,600
Collection Time: 250
Maximum Collection Count: 16
Keep Connection: 600
Maximum Commands: 5
Session Time Out: 45
Character Buffer Size: 512
Maximum Threads: 17
Lock Quota: 6,144
Lock Increment: 10
Maximum Locks: 500
Pipe Increment: 10
Maximum Pipes: 500
Cache Time Out: 40
Dormant File Limit: 45
Read Ahead Throughput: 4,294,967,295
Mailslot Buffers: 3
Server Announce Buffers: 20
Illegal Datagrams: 5
Datagram Reset Frequency: 60
Log Election Packets: False
Use Opportunistic Locking: True
Use Unlock Behind: True
Use Close Behind: True
Buffer Pipes: True
Use Lock, Read, Unlock: True
Use NT Caching: True
Use Raw Read: True
Use Raw Write: True
Use Write Raw Data: True
Use Encryption: True
Buffer Deny Write Files: True
Buffer Read Only Files: True
Force Core Creation: True
512 Byte Max Transfer: False
Bytes Received: 1,141
SMB's Received: 12
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Bytes Transmitted: 1,420
SMB's Transmitted: 12
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 6,720
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Initally Failed Operations: 0
Failed Completion Operations: 0
Read Operations: 0
Random Read Operations: 0
Read SMB's: 0
Large Read SMB's: 0
Small Read SMB's: 0
Write Operations: 112
Random Write Operations: 0
Write SMB's: 0
Large Write SMB's: 0
Small Write SMB's: 0
Raw Reads Denied: 0
Raw Writes Denied: 0
Network Errors: 0
Sessions: 1
Failed Sessions: 0
Reconnects: 0
Core Connects: 0
LM 2.0 Connects: 0
LM 2.x Connects: 0
Windows NT Connects: 1
Server Disconnects: 0
Hung Sessions: 0
Use Count: 0
Failed Use Count: 0
Current Commands: 0
Server File Opens: 929,734,398

Server Device Opens: 0
Server Jobs Queued: 589,824
Server Session Opens: 2
Server Sessions Timed Out: 2,147,483,736
Server Sessions Errored Out: 3
Server Password Errors: 2,147,483,776
Server Permission Errors: 4
Server System Errors: 2,147,483,856
Server Bytes Sent: 9,223,373,033,287,188,485
Server Bytes Received: 9,223,373,892,280,647,686
Server Average Response Time: 9
Server Request Buffers Needed: 2,147,484,640
Server Big Buffers Needed: 11

Disk Controller Configuration Parameters

Adapter 0

Adapter No: 0

Number of Logical Drives : 1

Logical Drive = 0
Span Depth = 5
Raid Level = 0,
Read Ahead = NORMAL
Stripe Size = 64KB,
Status = OPTIMAL
Write Policy = WRITE_THRU,
Direct IO = DIRECT_IO,
Number of Stripes = 8
SPAN Number = 0
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 0
Device Number = 1
Channel Number = 1
Target Number = 0
Device Number = 2
Channel Number = 2
Target Number = 0
Device Number = 3
Channel Number = 3
Target Number = 0
Device Number = 4
Channel Number = 0
Target Number = 1
Device Number = 5
Channel Number = 1
Target Number = 1
Device Number = 6
Channel Number = 2
Target Number = 1
Device Number = 7
Channel Number = 3
Target Number = 1
SPAN Number = 1
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 2
Device Number = 1
Channel Number = 1
Target Number = 2
Device Number = 2
Channel Number = 2
Target Number = 2
Device Number = 3
Channel Number = 3
Target Number = 2
Device Number = 4
Channel Number = 0
Target Number = 3
Device Number = 5
Channel Number = 1
Target Number = 3
Device Number = 6
Channel Number = 2
Target Number = 3
Device Number = 7
Channel Number = 3
Target Number = 3
SPAN Number = 2
Starting Block = 0
Number of blocks = 17770496
Device Number = 0

```
                Channel Number = 0
                Target Number = 4
        Device Number = 1
                Channel Number = 1
                Target Number = 4
        Device Number = 2
                Channel Number = 2
                Target Number = 4
        Device Number = 3
                Channel Number = 3
                Target Number = 4
        Device Number = 4
                Channel Number = 0
                Target Number = 8
        Device Number = 5
                Channel Number = 1
                Target Number = 8
        Device Number = 6
                Channel Number = 2
                Target Number = 8
        Device Number = 7
                Channel Number = 3
                Target Number = 8
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 9
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 9
        Device Number = 3
                Channel Number = 3
                Target Number = 9
        Device Number = 4
                Channel Number = 0
                Target Number = 10
        Device Number = 5
                Channel Number = 1
                Target Number = 10
        Device Number = 6
                Channel Number = 2
                Target Number = 10
        Device Number = 7
                Channel Number = 3
                Target Number = 10
SPAN Number = 4
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 11
        Device Number = 1
                Channel Number = 1
                Target Number = 11
        Device Number = 2
                Channel Number = 2
                Target Number = 11
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 0
                Target Number = 12
        Device Number = 5
                Channel Number = 1
                Target Number = 12
        Device Number = 6
                Channel Number = 2
                Target Number = 12
        Device Number = 7
                Channel Number = 3
                Target Number = 12

(Channel 0, ID 0)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 1)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks
```

(Channel 0, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

Adapter 1

Adapter No: 1

Number of Logical Drives : 1

Logical Drive = 0
Span Depth = 5
Raid Level = 0,
Read Ahead = NORMAL
Stripe Size = 64KB,
Status = OPTIMAL
Write Policy = WRITE_THRU,
Direct IO = DIRECT_IO,
Number of Stripes = 8
SPAN Number = 0
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 0
Device Number = 1
Channel Number = 1
Target Number = 0
Device Number = 2
Channel Number = 2
Target Number = 0
Device Number = 3

```
                Channel Number = 3
                Target Number = 0
        Device Number = 4
                Channel Number = 0
                Target Number = 1
        Device Number = 5
                Channel Number = 1
                Target Number = 1
        Device Number = 6
                Channel Number = 2
                Target Number = 1
        Device Number = 7
                Channel Number = 3
                Target Number = 1
SPAN Number = 1
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 2
        Device Number = 1
                Channel Number = 1
                Target Number = 2
        Device Number = 2
                Channel Number = 2
                Target Number = 2
        Device Number = 3
                Channel Number = 3
                Target Number = 2
        Device Number = 4
                Channel Number = 0
                Target Number = 3
        Device Number = 5
                Channel Number = 1
                Target Number = 3
        Device Number = 6
                Channel Number = 2
                Target Number = 3
        Device Number = 7
                Channel Number = 3
                Target Number = 3
SPAN Number = 2
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 4
        Device Number = 1
                Channel Number = 1
                Target Number = 4
        Device Number = 2
                Channel Number = 2
                Target Number = 4
        Device Number = 3
                Channel Number = 3
                Target Number = 4
        Device Number = 4
                Channel Number = 0
                Target Number = 8
        Device Number = 5
                Channel Number = 1
                Target Number = 8
        Device Number = 6
                Channel Number = 2
                Target Number = 8
        Device Number = 7
                Channel Number = 3
                Target Number = 8
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 9
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 9
        Device Number = 3
                Channel Number = 3
                Target Number = 9
        Device Number = 4
                Channel Number = 0
                Target Number = 10
        Device Number = 5
                Channel Number = 1
                Target Number = 10
        Device Number = 6
                Channel Number = 2
                Target Number = 10
        Device Number = 7
                Channel Number = 3
                Target Number = 10
SPAN Number = 4
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 11
        Device Number = 1
                Channel Number = 1
                Target Number = 11
        Device Number = 2
                Channel Number = 2
                Target Number = 11
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 0
```

Target Number = 12
Device Number = 5
Channel Number = 1
Target Number = 12
Device Number = 6
Channel Number = 2
Target Number = 12
Device Number = 7
Channel Number = 3
Target Number = 12

(Channel 0, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

Adapter 2

Adapter No: 2

Number of Logical Drives : 1

Logical Drive = 0
Span Depth = 5
Raid Level = 0,
Read Ahead = NORMAL
Stripe Size = 64KB,
Status = OPTIMAL
Write Policy = WRITE_THRU,
Direct IO = DIRECT_IO,
Number of Stripes = 8
SPAN Number = 0
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 0
Device Number = 1
Channel Number = 1
Target Number = 0
Device Number = 2
Channel Number = 2
Target Number = 0
Device Number = 3
Channel Number = 3
Target Number = 0
Device Number = 4
Channel Number = 0
Target Number = 1
Device Number = 5
Channel Number = 1
Target Number = 1
Device Number = 6
Channel Number = 2
Target Number = 1
Device Number = 7
Channel Number = 3
Target Number = 1
SPAN Number = 1
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 2
Device Number = 1
Channel Number = 1

```
                Target Number = 2
        Device Number = 2
                Channel Number = 2
                Target Number = 2
        Device Number = 3
                Channel Number = 3
                Target Number = 2
        Device Number = 4
                Channel Number = 0
                Target Number = 3
        Device Number = 5
                Channel Number = 1
                Target Number = 3
        Device Number = 6
                Channel Number = 2
                Target Number = 3
        Device Number = 7
                Channel Number = 3
                Target Number = 3
SPAN Number = 2
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 4
        Device Number = 1
                Channel Number = 1
                Target Number = 4
        Device Number = 2
                Channel Number = 2
                Target Number = 4
        Device Number = 3
                Channel Number = 3
                Target Number = 4
        Device Number = 4
                Channel Number = 0
                Target Number = 8
        Device Number = 5
                Channel Number = 1
                Target Number = 8
        Device Number = 6
                Channel Number = 2
                Target Number = 8
        Device Number = 7
                Channel Number = 3
                Target Number = 8
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 9
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 9
        Device Number = 3
                Channel Number = 3
                Target Number = 9
        Device Number = 4
                Channel Number = 0
                Target Number = 10
        Device Number = 5
                Channel Number = 1
                Target Number = 10
        Device Number = 6
                Channel Number = 2
                Target Number = 10
        Device Number = 7
                Channel Number = 3
                Target Number = 10
SPAN Number = 4
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 11
        Device Number = 1
                Channel Number = 1
                Target Number = 11
        Device Number = 2
                Channel Number = 2
                Target Number = 11
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 0
                Target Number = 12
        Device Number = 5
                Channel Number = 1
                Target Number = 12
        Device Number = 6
                Channel Number = 2
                Target Number = 12
        Device Number = 7
                Channel Number = 3
                Target Number = 12


(Channel 0, ID 0)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 1)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 2)
        Type = HARDDISK,      Current Status = ONLINE
```

```
        Size 17770496 blocks

(Channel 0, ID 3)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 4)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 8)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 9)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 10)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 11)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 12)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 0)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 1)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 2)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 3)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 4)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 8)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 9)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 10)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 11)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 12)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 2, ID 0)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 2, ID 1)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 2, ID 2)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 2, ID 3)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 2, ID 4)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 2, ID 8)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 2, ID 9)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 2, ID 10)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 2, ID 11)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 2, ID 12)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 3, ID 0)
        Type = HARDDISK,      Current Status = ONLINE
```

Size 17770496 blocks

(Channel 3, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

Adapter 3

Adapter No: 3

Number of Logical Drives : 1

Logical Drive = 0
Span Depth = 5
Raid Level = 0,
Read Ahead = NORMAL
Stripe Size = 64KB,
Status = OPTIMAL
Write Policy = WRITE_THRU,
Direct IO = DIRECT_IO,
Number of Stripes = 8
SPAN Number = 0
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 0
Device Number = 1
Channel Number = 1
Target Number = 0
Device Number = 2
Channel Number = 2
Target Number = 0
Device Number = 3
Channel Number = 3
Target Number = 0
Device Number = 4
Channel Number = 0
Target Number = 1
Device Number = 5
Channel Number = 1
Target Number = 1
Device Number = 6
Channel Number = 2
Target Number = 1
Device Number = 7
Channel Number = 3
Target Number = 1
SPAN Number = 1
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 2
Device Number = 1
Channel Number = 1
Target Number = 2
Device Number = 2
Channel Number = 2
Target Number = 2
Device Number = 3
Channel Number = 3
Target Number = 2
Device Number = 4
Channel Number = 0
Target Number = 3
Device Number = 5
Channel Number = 1
Target Number = 3
Device Number = 6
Channel Number = 2
Target Number = 3
Device Number = 7
Channel Number = 3
Target Number = 3
SPAN Number = 2
Starting Block = 0
Number of blocks = 17770496

```
        Device Number = 0
                Channel Number = 0
                Target Number = 4
        Device Number = 1
                Channel Number = 1
                Target Number = 4
        Device Number = 2
                Channel Number = 2
                Target Number = 4
        Device Number = 3
                Channel Number = 3
                Target Number = 4
        Device Number = 4
                Channel Number = 0
                Target Number = 8
        Device Number = 5
                Channel Number = 1
                Target Number = 8
        Device Number = 6
                Channel Number = 2
                Target Number = 8
        Device Number = 7
                Channel Number = 3
                Target Number = 8
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 9
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 9
        Device Number = 3
                Channel Number = 3
                Target Number = 9
        Device Number = 4
                Channel Number = 0
                Target Number = 10
        Device Number = 5
                Channel Number = 1
                Target Number = 10
        Device Number = 6
                Channel Number = 2
                Target Number = 10
        Device Number = 7
                Channel Number = 3
                Target Number = 10
SPAN Number = 4
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 11
        Device Number = 1
                Channel Number = 1
                Target Number = 11
        Device Number = 2
                Channel Number = 2
                Target Number = 11
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 0
                Target Number = 12
        Device Number = 5
                Channel Number = 1
                Target Number = 12
        Device Number = 6
                Channel Number = 2
                Target Number = 12
        Device Number = 7
                Channel Number = 3
                Target Number = 12


(Channel 0, ID 0)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 1)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 2)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 3)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 4)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 8)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 9)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 10)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks
```

(Channel 0, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

Adapter 4

Adapter No: 4

Number of Logical Drives : 1

Logical Drive = 0
Span Depth = 5
Raid Level = 0,
Read Ahead = NORMAL
Stripe Size = 64KB,
Status = OPTIMAL
Write Policy = WRITE_THRU,
Direct IO = DIRECT_IO,
Number of Stripes = 8
SPAN Number = 0
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 0
Device Number = 1
Channel Number = 1
Target Number = 0
Device Number = 2
Channel Number = 2
Target Number = 0
Device Number = 3
Channel Number = 3
Target Number = 0
Device Number = 4
Channel Number = 0
Target Number = 1
Device Number = 5
Channel Number = 1
Target Number = 1
Device Number = 6
Channel Number = 2
Target Number = 1
Device Number = 7
Channel Number = 3
Target Number = 1
SPAN Number = 1
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 2
Device Number = 1
Channel Number = 1
Target Number = 2
Device Number = 2
Channel Number = 2
Target Number = 2
Device Number = 3
Channel Number = 3
Target Number = 2
Device Number = 4
Channel Number = 0
Target Number = 3
Device Number = 5
Channel Number = 1
Target Number = 3
Device Number = 6
Channel Number = 2
Target Number = 3
Device Number = 7
Channel Number = 3
Target Number = 3
SPAN Number = 2
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 4
Device Number = 1
Channel Number = 1
Target Number = 4
Device Number = 2
Channel Number = 2
Target Number = 4
Device Number = 3
Channel Number = 3
Target Number = 4
Device Number = 4
Channel Number = 0
Target Number = 8
Device Number = 5
Channel Number = 1
Target Number = 8
Device Number = 6
Channel Number = 2
Target Number = 8
Device Number = 7

```
                Channel Number = 3
                Target Number = 8
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 9
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 9
        Device Number = 3
                Channel Number = 3
                Target Number = 9
        Device Number = 4
                Channel Number = 0
                Target Number = 10
        Device Number = 5
                Channel Number = 1
                Target Number = 10
        Device Number = 6
                Channel Number = 2
                Target Number = 10
        Device Number = 7
                Channel Number = 3
                Target Number = 10
SPAN Number = 4
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 11
        Device Number = 1
                Channel Number = 1
                Target Number = 11
        Device Number = 2
                Channel Number = 2
                Target Number = 11
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 0
                Target Number = 12
        Device Number = 5
                Channel Number = 1
                Target Number = 12
        Device Number = 6
                Channel Number = 2
                Target Number = 12
        Device Number = 7
                Channel Number = 3
                Target Number = 12
```

```
(Channel 0, ID 0)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 1)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 2)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 3)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 4)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 8)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 9)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 10)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 11)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 12)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 0)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 1)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 2)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks
```

(Channel 1, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

Adapter 5

Adapter No: 5

Number of Logical Drives : 1

Logical Drive = 0
Span Depth = 5
Raid Level = 0,
Read Ahead = NORMAL
Stripe Size = 64KB,
Status = OPTIMAL
Write Policy = WRITE_THRU,
Direct IO = DIRECT_IO,
Number of Stripes = 8
SPAN Number = 0
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 0
Device Number = 1
Channel Number = 1
Target Number = 0
Device Number = 2
Channel Number = 2
Target Number = 0
Device Number = 3
Channel Number = 3
Target Number = 0
Device Number = 4
Channel Number = 0
Target Number = 1
Device Number = 5
Channel Number = 1
Target Number = 1
Device Number = 6
Channel Number = 2
Target Number = 1
Device Number = 7
Channel Number = 3
Target Number = 1
SPAN Number = 1
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 2
Device Number = 1
Channel Number = 1
Target Number = 2
Device Number = 2
Channel Number = 2
Target Number = 2
Device Number = 3
Channel Number = 3
Target Number = 2
Device Number = 4
Channel Number = 0
Target Number = 3
Device Number = 5
Channel Number = 1
Target Number = 3
Device Number = 6
Channel Number = 2
Target Number = 3
Device Number = 7
Channel Number = 3
Target Number = 3
SPAN Number = 2
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 4
Device Number = 1
Channel Number = 1
Target Number = 4
Device Number = 2
Channel Number = 2
Target Number = 4
Device Number = 3
Channel Number = 3
Target Number = 4
Device Number = 4
Channel Number = 0
Target Number = 8

```
        Device Number = 5
                Channel Number = 1
                Target Number = 8
        Device Number = 6
                Channel Number = 2
                Target Number = 8
        Device Number = 7
                Channel Number = 3
                Target Number = 8
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 9
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 9
        Device Number = 3
                Channel Number = 3
                Target Number = 9
        Device Number = 4
                Channel Number = 0
                Target Number = 10
        Device Number = 5
                Channel Number = 1
                Target Number = 10
        Device Number = 6
                Channel Number = 2
                Target Number = 10
        Device Number = 7
                Channel Number = 3
                Target Number = 10
SPAN Number = 4
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 11
        Device Number = 1
                Channel Number = 1
                Target Number = 11
        Device Number = 2
                Channel Number = 2
                Target Number = 11
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 0
                Target Number = 12
        Device Number = 5
                Channel Number = 1
                Target Number = 12
        Device Number = 6
                Channel Number = 2
                Target Number = 12
        Device Number = 7
                Channel Number = 3
                Target Number = 12

(Channel 0, ID 0)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 1)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 2)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 3)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 4)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 8)
        Type = HARDDISK,    Current Status = ONLINE
        Size 17770496 blocks
```

(Channel 0, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

Adapter 6

Adapter No: 6

Number of Logical Drives : 1

Logical Drive = 0
Span Depth = 5
Raid Level = 0,
Read Ahead = NORMAL
Stripe Size = 64KB,
Status = OPTIMAL
Write Policy = WRITE_THRU,
Direct IO = DIRECT_IO,
Number of Stripes = 8
SPAN Number = 0
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 0
Device Number = 1
Channel Number = 1
Target Number = 0
Device Number = 2
Channel Number = 2
Target Number = 0
Device Number = 3
Channel Number = 3
Target Number = 0
Device Number = 4
Channel Number = 0
Target Number = 1
Device Number = 5
Channel Number = 1
Target Number = 1
Device Number = 6
Channel Number = 2
Target Number = 1

```
        Device Number = 7
                Channel Number = 3
                Target Number = 1
SPAN Number = 1
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 2
        Device Number = 1
                Channel Number = 1
                Target Number = 2
        Device Number = 2
                Channel Number = 2
                Target Number = 2
        Device Number = 3
                Channel Number = 3
                Target Number = 2
        Device Number = 4
                Channel Number = 0
                Target Number = 3
        Device Number = 5
                Channel Number = 1
                Target Number = 3
        Device Number = 6
                Channel Number = 2
                Target Number = 3
        Device Number = 7
                Channel Number = 3
                Target Number = 3
SPAN Number = 2
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 4
        Device Number = 1
                Channel Number = 1
                Target Number = 4
        Device Number = 2
                Channel Number = 2
                Target Number = 4
        Device Number = 3
                Channel Number = 3
                Target Number = 4
        Device Number = 4
                Channel Number = 0
                Target Number = 8
        Device Number = 5
                Channel Number = 1
                Target Number = 8
        Device Number = 6
                Channel Number = 2
                Target Number = 8
        Device Number = 7
                Channel Number = 3
                Target Number = 8
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 9
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 9
        Device Number = 3
                Channel Number = 3
                Target Number = 9
        Device Number = 4
                Channel Number = 0
                Target Number = 10
        Device Number = 5
                Channel Number = 1
                Target Number = 10
        Device Number = 6
                Channel Number = 2
                Target Number = 10
        Device Number = 7
                Channel Number = 3
                Target Number = 10
SPAN Number = 4
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
```

```
                    Channel Number = 0
                    Target Number = 11
          Device Number = 1
                    Channel Number = 1
                    Target Number = 11
          Device Number = 2
                    Channel Number = 2
                    Target Number = 11
          Device Number = 3
                    Channel Number = 3
                    Target Number = 11
          Device Number = 4
                    Channel Number = 0
                    Target Number = 12
          Device Number = 5
                    Channel Number = 1
                    Target Number = 12
          Device Number = 6
                    Channel Number = 2
                    Target Number = 12
          Device Number = 7
                    Channel Number = 3
                    Target Number = 12

(Channel 0, ID 0)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 0, ID 1)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 0, ID 2)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 0, ID 3)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 0, ID 4)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 0, ID 8)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 0, ID 9)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 0, ID 10)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 0, ID 11)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 0, ID 12)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 1, ID 0)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 1, ID 1)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 1, ID 2)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 1, ID 3)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 1, ID 4)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks

(Channel 1, ID 8)
          Type = HARDDISK,      Current Status = ONLINE
          Size 17770496 blocks
```

(Channel 1, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

Adapter 7

Adapter No: 7

Number of Logical Drives : 1

Logical Drive = 0
Span Depth = 5
Raid Level = 0,
Read Ahead = NORMAL
Stripe Size = 64KB,
Status = OPTIMAL
Write Policy = WRITE_THRU,
Direct IO = DIRECT_IO,
Number of Stripes = 8
SPAN Number = 0
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 0
Device Number = 1
Channel Number = 1
Target Number = 0
Device Number = 2
Channel Number = 2
Target Number = 0
Device Number = 3
Channel Number = 3
Target Number = 0
Device Number = 4
Channel Number = 0
Target Number = 1
Device Number = 5
Channel Number = 1
Target Number = 1
Device Number = 6
Channel Number = 2
Target Number = 1
Device Number = 7
Channel Number = 3
Target Number = 1
SPAN Number = 1
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 2
Device Number = 1
Channel Number = 1
Target Number = 2
Device Number = 2
Channel Number = 2
Target Number = 2
Device Number = 3
Channel Number = 3
Target Number = 2
Device Number = 4
Channel Number = 0
Target Number = 3
Device Number = 5
Channel Number = 1
Target Number = 3
Device Number = 6
Channel Number = 2
Target Number = 3
Device Number = 7
Channel Number = 3
Target Number = 3
SPAN Number = 2
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 4
Device Number = 1
Channel Number = 1
Target Number = 4
Device Number = 2

```
                Channel Number = 2
                Target Number = 4
        Device Number = 3
                Channel Number = 3
                Target Number = 4
        Device Number = 4
                Channel Number = 0
                Target Number = 8
        Device Number = 5
                Channel Number = 1
                Target Number = 8
        Device Number = 6
                Channel Number = 2
                Target Number = 8
        Device Number = 7
                Channel Number = 3
                Target Number = 8
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 9
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 9
        Device Number = 3
                Channel Number = 3
                Target Number = 9
        Device Number = 4
                Channel Number = 0
                Target Number = 10
        Device Number = 5
                Channel Number = 1
                Target Number = 10
        Device Number = 6
                Channel Number = 2
                Target Number = 10
        Device Number = 7
                Channel Number = 3
                Target Number = 10
SPAN Number = 4
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 11
        Device Number = 1
                Channel Number = 1
                Target Number = 11
        Device Number = 2
                Channel Number = 2
                Target Number = 11
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 0
                Target Number = 12
        Device Number = 5
                Channel Number = 1
                Target Number = 12
        Device Number = 6
                Channel Number = 2
                Target Number = 12
        Device Number = 7
                Channel Number = 3
                Target Number = 12

(Channel 0, ID 0)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 1)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 2)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 3)
```

Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 4)

Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 8)

Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

Adapter 8

Adapter No: 8

Number of Logical Drives : 1

Logical Drive = 0
Span Depth = 5
Raid Level = 0,
Read Ahead = NORMAL
Stripe Size = 64KB,
Status = OPTIMAL
Write Policy = WRITE_THRU,
Direct IO = DIRECT_IO,
Number of Stripes = 8
SPAN Number = 0
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 0
Device Number = 1
Channel Number = 1
Target Number = 0
Device Number = 2
Channel Number = 2
Target Number = 0
Device Number = 3

```
                Channel Number = 3
                Target Number = 0
        Device Number = 4
                Channel Number = 0
                Target Number = 1
        Device Number = 5
                Channel Number = 1
                Target Number = 1
        Device Number = 6
                Channel Number = 2
                Target Number = 1
        Device Number = 7
                Channel Number = 3
                Target Number = 1
SPAN Number = 1
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 2
        Device Number = 1
                Channel Number = 1
                Target Number = 2
        Device Number = 2
                Channel Number = 2
                Target Number = 2
        Device Number = 3
                Channel Number = 3
                Target Number = 2
        Device Number = 4
                Channel Number = 0
                Target Number = 3
        Device Number = 5
                Channel Number = 1
                Target Number = 3
        Device Number = 6
                Channel Number = 2
                Target Number = 3
        Device Number = 7
                Channel Number = 3
                Target Number = 3
SPAN Number = 2
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 4
        Device Number = 1
                Channel Number = 1
                Target Number = 4
        Device Number = 2
                Channel Number = 2
                Target Number = 4
        Device Number = 3
                Channel Number = 3
                Target Number = 4
        Device Number = 4
                Channel Number = 0
                Target Number = 8
        Device Number = 5
                Channel Number = 1
                Target Number = 8
        Device Number = 6
                Channel Number = 2
                Target Number = 8
        Device Number = 7
                Channel Number = 3
                Target Number = 8
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 9
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 9
        Device Number = 3
                Channel Number = 3
                Target Number = 9
        Device Number = 4
                Channel Number = 0
                Target Number = 10
        Device Number = 5
                Channel Number = 1
```

```
                Target Number = 10
        Device Number = 6
                Channel Number = 2
                Target Number = 10
        Device Number = 7
                Channel Number = 3
                Target Number = 10
SPAN Number = 4
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 11
        Device Number = 1
                Channel Number = 1
                Target Number = 11
        Device Number = 2
                Channel Number = 2
                Target Number = 11
        Device Number = 3
                Channel Number = 3
                Target Number = 11
        Device Number = 4
                Channel Number = 0
                Target Number = 12
        Device Number = 5
                Channel Number = 1
                Target Number = 12
        Device Number = 6
                Channel Number = 2
                Target Number = 12
        Device Number = 7
                Channel Number = 3
                Target Number = 12
```

```
(Channel 0, ID 0)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 1)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 2)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 3)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 4)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 8)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 9)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 10)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 11)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 0, ID 12)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 0)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 1)
        Type = HARDDISK,      Current Status = ONLINE
        Size 17770496 blocks

(Channel 1, ID 2)
        Type = HARDDISK,      Current Status = ONLINE
```

Size 17770496 blocks

(Channel 1, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 4)
Type = HARDDISK, Current Status = ONLINE

Size 17770496 blocks

(Channel 3, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

Adapter 9

Adapter No: 9

Number of Logical Drives : 1

Logical Drive = 0
Span Depth = 5
Raid Level = 0,
Read Ahead = NORMAL
Stripe Size = 64KB,
Status = OPTIMAL
Write Policy = WRITE_THRU,
Direct IO = DIRECT_IO,
Number of Stripes = 8
SPAN Number = 0
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 0
Device Number = 1
Channel Number = 1
Target Number = 0
Device Number = 2
Channel Number = 2
Target Number = 0
Device Number = 3
Channel Number = 3
Target Number = 0
Device Number = 4
Channel Number = 0
Target Number = 1
Device Number = 5
Channel Number = 1
Target Number = 1
Device Number = 6
Channel Number = 2
Target Number = 1
Device Number = 7
Channel Number = 3
Target Number = 1
SPAN Number = 1
Starting Block = 0
Number of blocks = 17770496
Device Number = 0
Channel Number = 0
Target Number = 2
Device Number = 1
Channel Number = 1
Target Number = 2
Device Number = 2
Channel Number = 2
Target Number = 2
Device Number = 3
Channel Number = 3
Target Number = 2
Device Number = 4
Channel Number = 0
Target Number = 3
Device Number = 5
Channel Number = 1
Target Number = 3
Device Number = 6
Channel Number = 2

```
                Target Number = 3
        Device Number = 7
                Channel Number = 3
                Target Number = 3
SPAN Number = 2
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 4
        Device Number = 1
                Channel Number = 1
                Target Number = 4
        Device Number = 2
                Channel Number = 2
                Target Number = 4
        Device Number = 3
                Channel Number = 3
                Target Number = 4
        Device Number = 4
                Channel Number = 0
                Target Number = 8
        Device Number = 5
                Channel Number = 1
                Target Number = 8
        Device Number = 6
                Channel Number = 2
                Target Number = 8
        Device Number = 7
                Channel Number = 3
                Target Number = 8
SPAN Number = 3
        Starting Block = 0
        Number of blocks = 17770496
        Device Number = 0
                Channel Number = 0
                Target Number = 9
        Device Number = 1
                Channel Number = 1
                Target Number = 9
        Device Number = 2
                Channel Number = 2
                Target Number = 9
        Device Number = 3
                Channel Number = 3
                Target Number = 9
        Device Number = 4
                Channel Number = 0
                Target Number = 10
        Device Number = 5
                Channel Number = 1
                Target Number = 10
        Device Number = 6
                Channel Number = 2
                Target Number = 10
        Device Number = 7

Channel Number = 3
                Target Number = 10
        SPAN Number = 4
                Starting Block = 0
                Number of blocks = 17770496
                Device Number = 0
                        Channel Number = 0
                        Target Number = 11
                Device Number = 1
                        Channel Number = 1
                        Target Number = 11
                Device Number = 2
                        Channel Number = 2
                        Target Number = 11
                Device Number = 3
                        Channel Number = 3
                        Target Number = 11
                Device Number = 4
                        Channel Number = 0
                        Target Number = 12
                Device Number = 5
                        Channel Number = 1
                        Target Number = 12
                Device Number = 6
                        Channel Number = 2
                        Target Number = 12
                Device Number = 7
                        Channel Number = 3
                        Target Number = 12
```

Size 17770496 blocks

(Channel 0, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 0, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 1, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 1)
Type = HARDDISK, Current Status = ONLINE

Size 17770496 blocks

(Channel 2, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 2, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 0)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 1)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 2)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 3)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 4)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 8)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 9)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 10)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 11)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

(Channel 3, ID 12)
Type = HARDDISK, Current Status = ONLINE
Size 17770496 blocks

Netfinity Fibre Channel PCI Adapter

Profile for IBMSERV3_001

Profile Generated By: SYMplicity Storage Manager 06.22.25.15

Host: IBMSERV3

Controllers:

Name	Serial Number	Mode	Logical Units
A (Drive8)	1T84710324	Active	1
B	1T85910067	Offline	0

Number of Drives = 12

Detailed Controller Information for IBMSERV3_001

Parameters	Controller A (Drive8)	Controller B
Board Name:	4th Generation	
Board ID:	4766	
Board Serial Number:	1T84710324	1T85910067
Product ID:	3526	
Product Serial Number:	1T84710324	
Vendor ID:	IBM	
Date of Manufacture:	12/17/98	
Product Revision:	0301	
Host Interface:	Fibre Channel - FCP	Fibre Channel - FCP
Topology:	Arbitrated Loop - Private	Unknown
Controller ID - NL_Port (hex):	0000E0	000000
Preferred Address - AL_PA (hex):	E0	00
Preferred Loop ID (dec):	5	0
Host-side ID (dec):	5	0
Worldwide Port Name (hex):	200200A0B805A705	0000000000000000
Physical Connection:	Unknown	Unknown
Link Speed (MB/sec):	100	0
Boot Level:	03.01.00.00	
Firmware Level:	03.01.01.00	
Fibre Channel Level:	03.01.01.00	
Cache/Processor Size:	128/32 MB	128/0 MB

Drives:

Detailed Drive Information for IBMSERV3_001

Location	Capacity (MB)	Status	Vendor	Product ID	
[1,0]	17357	Optimal	IBM-PSG	ST318203LC	!#
[2,0]	17357	Optimal	IBM-PSG	ST318203LC	!#
[3,0]	17357	Optimal	IBM-PSG	ST318203LC	!#
[4,0]	17357	Optimal	IBM-PSG	ST318203LC	!#
[1,1]	17357	Optimal	IBM-PSG	ST318203LC	!#
[2,1]	17357	Optimal	IBM-PSG	ST318203LC	!#
[3,1]	17357	Optimal	IBM-PSG	ST318203LC	!#
[4,1]	17357	Optimal	IBM-PSG	ST318203LC	!#
[1,2]	17357	Optimal	IBM-PSG	ST318203LC	!#
[2,2]	17357	Optimal	IBM-PSG	ST318203LC	!#
[3,2]	17357	Optimal	IBM-PSG	ST318203LC	!#
[4,2]	17357	Optimal	IBM-PSG	ST318203LC	!#

Detailed Drive Information for IBMSERV3_001 (continued)

Location	Firmware Version	Serial Number	Date Code
[1,0]	B222	LR23757505	080499
[2,0]	B222	LR22476605	080499
[3,0]	B222	LR23701505	080499
[4,0]	B222	LR23221505	080499
[1,1]	B222	LR23577105	080499
[2,1]	B222	LR23547205	080499
[3,1]	B222	LR23598405	080499
[4,1]	B222	LR23592005	080499
[1,2]	B222	LR23716305	080499
[2,2]	B222	LR23512905	080499
[3,2]	B222	LR23304705	080499
[4,2]	B222	LR23267005	080399

Logical Units (LUNs):

Detailed LUN Information for IBMSERV3_001

LUN	Controller	Capacity (MB)	RAID Level
0	Drive8	103905	1

Detailed LUN Information for IBMSERV3_001 (continued)

LUN	Associated Drives
0	[1,0] [2,0] [3,0] [4,0] [1,1] [2,1] [3,1] [4,1] [1,2] [2,2] [3,2] [4,2]

Detailed LUN Information for IBMSERV3_001 (continued)

LUN	Segment Size (Blocks)	Write Cache	Cache Mirroring	Cache Without Batteries	Status
0	128	Disabled	Disabled	Disabled	Optimal

Drive Group Information for IBMSERV3_001

Group	No. of LUNS	RAID Level	No. of Drives	Total Capacity (MB)	Remaining Capacity (MB)
1	1	1	12	103905	0

Logical Unit Information for IBMSERV3_001

LUN	Device Group	RAID Name	Level	Cap. (MB)	Status
0	1	Drive8	1	103905	Optimal

Microsoft Windows 2000 Configuration Parameters

System Information report written at: 02/23/2000 11:20:57 AM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2128 Build 2128
OS Manufacturer	Microsoft Corporation
System Name	SQLCLIENT40
System Manufacturer	IBM
System Model	Netfinity 5000
System Type	X86-based PC
Processor	x86 Family 6 Model 7 Stepping 3 GenuineIntel ~600 Mhz
BIOS Version	IBM BIOS Ver 18.0
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	SQLCLIENT40\Administrator
Time Zone	Eastern Standard Time
Total Physical Memory	523,804 KB
Available Physical Memory	256,860 KB
Total Virtual Memory	1,802,692 KB
Available Virtual Memory	1,290,196 KB
Page File Space	1,278,888 KB
Page File	Not Available

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
IRQ 9	Microsoft ACPI-Compliant System
IRQ 9	S3 Trio64V2
IRQ 18	IBM 10/100 EtherJet PCI Adapter #2
IRQ 18	Standard OpenHCD USB Host Controller
IRQ 16	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
IRQ 16	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
IRQ 19	Intel(R) PRO/100+ Dual Port Server Adapter
IRQ 19	Intel(R) PRO/100+ Dual Port Server Adapter #2
IRQ 19	Intel(R) PRO/100+ Dual Port Server Adapter #3
IRQ 19	Intel(R) PRO/100+ Dual Port Server Adapter #4

[DMA]

Channel	Device	Status
2	Standard floppy disk controller	OK
4	Direct memory access controller	OK

[Forced Hardware]

Device PNP Device ID
No Forced Hardware

[I/O]

Address Range	Device	Status
0x0000-0x0CF7	PCI bus	OK
0x0000-0x0CF7	Direct memory access controller	OK
0x0D00-0x4AEB	PCI bus	OK
0x82E8-0xFFFF	PCI bus	OK
0x82E8-0xFFFF	S3 Trio64V2	OK
0x2000-0x201F	IBM 10/100 EtherJet PCI Adapter #2	OK
0x2200-0x22FF	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller	OK
0x2300-0x23FF	Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller	OK
0x2020-0x203F	IBM Netfinity Fault Tolerance PCI Adapter	OK
0x03C0-0x03CF	S3 Trio64V2	OK
0x03D4-0x03DB	S3 Trio64V2	OK
0x42E8-0x42E9	S3 Trio64V2	OK
0x4AE8-0x4AE9	S3 Trio64V2	OK
0x86E8-0x86EB	S3 Trio64V2	OK
0x8AE8-0x8AEB	S3 Trio64V2	OK
0x8EE8-0x8EEB	S3 Trio64V2	OK
0x92E8-0x92EB	S3 Trio64V2	OK

0x96E8-0x96EB S3 Trio64V2 OK
0x9AE8-0x9AEB S3 Trio64V2 OK
0x9EE8-0x9EEB S3 Trio64V2 OK
0xA2E8-0xA2EB S3 Trio64V2 OK
0xA6E8-0xA6EB S3 Trio64V2 OK
0xAAE8-0xAAEB S3 Trio64V2 OK
0xAEE8-0xAEEB S3 Trio64V2 OK
0xB6E8-0xB6EB S3 Trio64V2 OK
0xBAE8-0xBAEB S3 Trio64V2 OK
0xBEE8-0xBEEB S3 Trio64V2 OK
0xE2E8-0xE2EB S3 Trio64V2 OK
0xC2E8-0xC2EB S3 Trio64V2 OK
0xC6E8-0xC6EB S3 Trio64V2 OK
0xCAE8-0xCAEB S3 Trio64V2 OK
0xCEE8-0xCEEB S3 Trio64V2 OK
0xD2E8-0xD2EB S3 Trio64V2 OK
0xD6E8-0xD6EB S3 Trio64V2 OK
0xDAE8-0xDAEB S3 Trio64V2 OK
0xDEE8-0xDEEB S3 Trio64V2 OK
0xE6E8-0xE6EB S3 Trio64V2 OK
0xEAE8-0xEAEB S3 Trio64V2 OK
0xEEE8-0xEEEB S3 Trio64V2 OK
0xF6E8-0xF6EB S3 Trio64V2 OK
0xFAE8-0xFAEB S3 Trio64V2 OK
0xFEE8-0xFEEB S3 Trio64V2 OK
0x0A79-0x0A79 ISAPNP Read Data Port OK
0x0279-0x0279 ISAPNP Read Data Port OK
0x02F4-0x02F7 ISAPNP Read Data Port OK
0x0060-0x0060 Standard 101/102-Key or Microsoft Natural PS/2 Keyboard OK
0x0064-0x0064 Standard 101/102-Key or Microsoft Natural PS/2 Keyboard OK
0x03F0-0x03F5 Standard floppy disk controller OK
0x03F7-0x03F7 Standard floppy disk controller OK
0x0378-0x037F Printer Port (LPT1) OK
0x03F8-0x03FF Communications Port (COM1) OK
0x02F8-0x02FF Communications Port (COM2) OK
0x0020-0x0021 Advanced programmable interrupt controller OK
0x00A0-0x00A1 Advanced programmable interrupt controller OK
0x04D0-0x04D1 Advanced programmable interrupt controller OK
0x0080-0x008F Direct memory access controller OK
0x00C0-0x00DF Direct memory access controller OK
0x0040-0x0043 System timer OK
0x0070-0x0073 System CMOS/real time clock OK
0x0061-0x0061 System speaker OK
0x00F0-0x00FF Numeric data processor OK
0x0F50-0x0F58 Motherboard resources OK
0xFD00-0xFD3F Motherboard resources OK
0xFE80-0xFEBF Motherboard resources OK
0xFEC0-0xFEDF Motherboard resources OK
0xFFA0-0xFFAF Standard Dual Channel PCI IDE Controller OK
0x01F0-0x01F7 Primary IDE Channel OK
0x03F6-0x03F6 Primary IDE Channel OK
0x0170-0x0177 Secondary IDE Channel OK
0x0376-0x0376 Secondary IDE Channel OK
0x4AEC-0x82E7 PCI bus OK
0x6000-0x6FFF DEC 21152 PCI to PCI bridge OK
0x6100-0x611F Intel(R) PRO/100+ Dual Port Server Adapter OK
0x6120-0x613F Intel(R) PRO/100+ Dual Port Server Adapter #2 OK
0x5000-0x5FFF DEC 21152 PCI to PCI bridge OK
0x5100-0x511F Intel(R) PRO/100+ Dual Port Server Adapter #3 OK
0x5120-0x513F Intel(R) PRO/100+ Dual Port Server Adapter #4 OK
0x4B00-0x4B1F IBM 10/100 EtherJet PCI Adapter OK

[IRQs]

IRQ Number Device
9 Microsoft ACPI-Compliant System
9 S3 Trio64V2
18 IBM 10/100 EtherJet PCI Adapter #2
18 Standard OpenHCD USB Host Controller
16 Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
16 Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
17 IBM Netfinity Fault Tolerance PCI Adapter
1 Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12 PS/2 Compatible Mouse
6 Standard floppy disk controller
4 Communications Port (COM1)
3 Communications Port (COM2)
0 System timer
8 System CMOS/real time clock
13 Numeric data processor
14 Primary IDE Channel
19 Intel(R) PRO/100+ Dual Port Server Adapter
19 Intel(R) PRO/100+ Dual Port Server Adapter #2
19 Intel(R) PRO/100+ Dual Port Server Adapter #3
19 Intel(R) PRO/100+ Dual Port Server Adapter #4
22 IBM 10/100 EtherJet PCI Adapter

[Memory]

Range Device Status
0xC8000-0xCFFFF PCI bus OK
0xC8000-0xCFFFF System board OK
0xC0FF0000-0xFEBFFFFF PCI bus OK
0xFED00000-0xFEDFFFFF PCI bus OK
0xFEF00000-0xFFDFFFFF PCI bus OK
0xFEBFF000-0xFEBFFFFF IBM 10/100 EtherJet PCI Adapter #2 OK
0xFEA00000-0xFEAFFFFF IBM 10/100 EtherJet PCI Adapter #2 OK
0xFEBFE000-0xFEBFEFFF Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller OK
0xFEBFD000-0xFEBFDFFF Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller OK
0xFEBFCC00-0xFEBFCC1F IBM Netfinity Fault Tolerance PCI Adapter OK
0xF8000000-0xFBFFFFFF S3 Trio64V2 OK
0xA0000-0xAFFFF S3 Trio64V2 OK
0xFEC00000-0xFEC0FFFF Advanced programmable interrupt controller OK
0xFF700000-0xFF700FFF Standard OpenHCD USB Host Controller OK
0xD0000-0xDFFFF PCI bus OK
0x20000000-0xC0FEFFFF PCI bus OK
0xBD000000-0xBEAFFFFF DEC 21152 PCI to PCI bridge OK
0x61000000-0x617FFFFF DEC 21152 PCI to PCI bridge OK
0xBEA00000-0xBEA00FFF Intel(R) PRO/100+ Dual Port Server Adapter OK
0xBE800000-0xBE8FFFFF Intel(R) PRO/100+ Dual Port Server Adapter OK

0xBEA01000-0xBEA01FFF	Intel(R) PRO/100+ Dual Port Server Adapter #2	OK	
0xBE900000-0xBE9FFFFF	Intel(R) PRO/100+ Dual Port Server Adapter #2	OK	
0xBF000000-0xC0AFFFFF	DEC 21152 PCI to PCI bridge	OK	
0x61800000-0x61FFFFFF	DEC 21152 PCI to PCI bridge	OK	
0xC0A00000-0xC0A00FFF	Intel(R) PRO/100+ Dual Port Server Adapter #3	OK	
0xC0800000-0xC08FFFFF	Intel(R) PRO/100+ Dual Port Server Adapter #3	OK	
0xC0A01000-0xC0A01FFF	Intel(R) PRO/100+ Dual Port Server Adapter #4	OK	
0xC0900000-0xC09FFFFF	Intel(R) PRO/100+ Dual Port Server Adapter #4	OK	
0xC0FDF000-0xC0FDFFFF	IBM 10/100 EtherJet PCI Adapter	OK	
0xC0E00000-0xC0EFFFFF	IBM 10/100 EtherJet PCI Adapter	OK	
0x0000-0x9FFF	System board	OK	
0x100000-0xFFFFFF	System board	OK	
0x1000000-0x1FFFFFFF	System board	OK	
0xE0000-0xFFFFF	System board	OK	
0xFFFE0000-0xFFFFFFFF	System board	OK	
0xCC000-0xCFFFF	System board	OK	

[Components]

[Following are sub-categories of this main category]

[Multimedia]

[Following are sub-categories of this main category]

[Audio Codecs]

Codec	Manufacturer	Description	Status	File	Version	Size	Creation Date
c:\winnt\system32\iac25_32.ax	Intel Corporation	Indeo® audio software	OK				
C:\WINNT\System32\IAC25_32.AX	2.05.53	195.00 KB (199,680 bytes)					9/9/1999 8:00:00 PM
c:\winnt\system32\msg723.acm	Microsoft Corporation		OK				
C:\WINNT\System32\MSG723.ACM	4.4.3385	106.77 KB (109,328 bytes)					2/8/2000 3:25:38 PM
c:\winnt\system32\lhacm.acm	Microsoft Corporation		OK				
C:\WINNT\System32\LHACM.ACM	4.4.3385	33.27 KB (34,064 bytes)					2/8/2000 3:25:39 PM
c:\winnt\system32\tssoft32.acm	DSP GROUP, INC.		OK				
C:\WINNT\System32\TSSOFT32.ACM	1.01	9.27 KB (9,488 bytes)					9/9/1999 8:00:00 PM
c:\winnt\system32\msgsm32.acm	Microsoft Corporation		OK				
C:\WINNT\System32\MSGSM32.ACM	5.00.2113.1	22.27 KB (22,800 bytes)					9/9/1999 8:00:00 PM
c:\winnt\system32\msg711.acm	Microsoft Corporation		OK				
C:\WINNT\System32\MSG711.ACM	5.00.2113.1	10.27 KB (10,512 bytes)					9/9/1999 8:00:00 PM
c:\winnt\system32\imaadp32.acm	Microsoft Corporation		OK				
C:\WINNT\System32\IMAADP32.ACM	5.00.2113.1	16.27 KB (16,656 bytes)					9/9/1999 8:00:00 PM
c:\winnt\system32\msadp32.acm	Microsoft Corporation		OK				
C:\WINNT\System32\MSADP32.ACM	5.00.2113.1	14.77 KB (15,120 bytes)					9/9/1999 8:00:00 PM

[Video Codecs]

Codec	Manufacturer	Description	Status	File	Version	Size	Creation Date
c:\winnt\system32\ir50_32.dll	Intel Corporation	Indeo® video 5.10	OK				
C:\WINNT\System32\IR50_32.DLL	R.5.10.15.2.54	737.50 KB (755,200 bytes)					9/9/1999 8:00:00 PM
c:\winnt\system32\msh261.drv	Microsoft Corporation		OK				
C:\WINNT\System32\MSH261.DRV	4.4.3385	163.77 KB (167,696 bytes)					2/8/2000 3:25:39 PM
c:\winnt\system32\msh263.drv	Microsoft Corporation		OK				
C:\WINNT\System32\MSH263.DRV	4.4.3385	252.27 KB (258,320 bytes)					2/8/2000 3:24:49 PM
c:\winnt\system32\msvidc32.dll	Microsoft Corporation		OK				
C:\WINNT\System32\MSVIDC32.DLL	5.00.2113.1	27.27 KB (27,920 bytes)					9/9/1999 8:00:00 PM
c:\winnt\system32\msrle32.dll	Microsoft Corporation		OK				
C:\WINNT\System32\MSRLE32.DLL	5.00.2113.1	10.77 KB (11,024 bytes)					9/9/1999 8:00:00 PM
c:\winnt\system32\ir32_32.dll	Intel(R) Corporation		OK				
C:\WINNT\System32\IR32_32.DLL	Not Available	194.50 KB (199,168 bytes)					9/9/1999 8:00:00 PM
c:\winnt\system32\iccvid.dll	Radius Inc.	OK	C:\WINNT\System32\ICCVID.DLL	1.10.0.6	108.00 KB (110,592 bytes)		9/9/1999 8:00:00 PM

[CD-ROM]

Item	Value
Drive	D:
Description	CD-ROM Drive
Media Loaded	False
Media Type	CD-ROM
Name	SANYO CD-ROM CRD-1332P
Manufacturer	(Standard CD-ROM drives)
Status	Unknown
Transfer Rate	Not Available
SCSI Target ID	0
PNP Device ID	IDE\CDROMSANYO_CD-ROM_CRD-1332P_____1.03____\5&CCFD636&0&0.0.0

[Sound Device]

Item	Value
No sound devices	

[Display]

Item	Value
Name	S3 Trio64V2
PNP Device ID	PCI\VEN_5333&DEV_8901&SUBSYS_00000000&REV_16\3&267A616A&0&50
Adapter Type	S3 TrioV2 / SGRAM memory, S3 compatible
Adapter Description	S3 Trio64V2
Adapter RAM	1024.00 KB (1,048,576 bytes)
Installed Drivers	s3mtrio.dll,s3mvirge.dll
Driver Version	5.1024.329.0002
INF File	display.inf (s3m section)
Color Planes	1
Color Table Entries	65536
Resolution	800 x 600 x 60 hertz
Bits/Pixel	16

[Infrared]

Item	Value
No infrared devices	

[Input]

[Following are sub-categories of this main category]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\4&3D912C8&0
NumberOfFunctionKeys 12

[Pointing Device]

Item Value
Hardware Type PS/2 Compatible Mouse
Number of Buttons 3
Status OK
PNP Device ID ACPI\PNP0F13\4&3D912C8&0
Power Management Supported False
Double Click Threshold 6
Handedness Right Handed Operation

[Modem]

Item Value
No modems

[Network]

[Following are sub-categories of this main category]

[Adapter]

Item Value
Name [000] IBM Netfinity Fault Tolerance PCI Adapter
Adapter Type Ethernet 802.3
Product Name IBM Netfinity Fault Tolerance PCI Adapter
Installed True
PNP Device ID PCI\VEN_1022&DEV_2000&SUBSYS_20001014&REV_33\3&267A616A&0&48
Last Reset 2/22/2000 12:35:01 PM
Index 0
Service Name PCNet5
IP Address 192.168.125.40
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:04:AC:B8:16:50
Service Name PCnet
IRQ Number 17
I/O Port 0x2020-0x203F
Driver 28.77 KB (29,456 bytes)

Name [001] IBM 10/100 EtherJet PCI Adapter
Adapter Type Ethernet 802.3
Product Name IBM 10/100 EtherJet PCI Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_005C1014&REV_05\3&13C0B0C5&0&20
Last Reset 2/22/2000 12:35:01 PM
Index 1
Service Name IBMFE
IP Address 192.168.46.1
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:04:AC:D8:06:93
Service Name IBMFE
IRQ Number 22
I/O Port 0x4B00-0x4B1F
Driver 78.27 KB (80,144 bytes)

Name [002] IBM 10/100 EtherJet PCI Adapter
Adapter Type Ethernet 802.3
Product Name IBM 10/100 EtherJet PCI Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_005C1014&REV_05\3&267A616A&0&08
Last Reset 2/22/2000 12:35:01 PM
Index 2
Service Name IBMFE
IP Address 192.168.45.1
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:04:AC:D8:1A:10
Service Name IBMFE
IRQ Number 18
I/O Port 0x2000-0x201F
Driver 78.27 KB (80,144 bytes)

Name [003] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/100+ Dual Port Server Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_10F08086&REV_05\4&28315C84&0&2010
Last Reset 2/22/2000 12:35:01 PM
Index 3
Service Name E100B
IP Address 192.168.41.1
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:90:27:E8:71:08
Service Name E100B

IRQ Number 19
I/O Port 0x6100-0x611F
Driver 78.27 KB (80,144 bytes)

Name [004] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/100+ Dual Port Server Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_10F08086&REV_05\4&28315C84&0&2810
Last Reset 2/22/2000 12:35:01 PM
Index 4
Service Name E100B
IP Address 192.168.42.1
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:90:27:E8:71:09
Service Name E100B
IRQ Number 19
I/O Port 0x6120-0x613F
Driver 78.27 KB (80,144 bytes)

Name [005] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/100+ Dual Port Server Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_10F08086&REV_05\4&666544D&0&2018
Last Reset 2/22/2000 12:35:01 PM
Index 5
Service Name E100B
IP Address 192.168.43.1
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:90:27:E2:44:2F
Service Name E100B
IRQ Number 19
I/O Port 0x5100-0x511F
Driver 78.27 KB (80,144 bytes)

Name [006] Intel(R) PRO/100+ Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/100+ Dual Port Server Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_10F08086&REV_05\4&666544D&0&2818
Last Reset 2/22/2000 12:35:01 PM
Index 6
Service Name E100B
IP Address 192.168.44.1
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:90:27:E2:44:30
Service Name E100B
IRQ Number 19
I/O Port 0x5120-0x513F
Driver 78.27 KB (80,144 bytes)

Name [007] RAS Async Adapter
Adapter Type RAS Async Adapter
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 2/22/2000 12:35:01 PM
Index 7
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available

Name [008] WAN Miniport (L2TP)
Adapter Type WAN Miniport (L2TP)
Product Name WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 2/22/2000 12:35:01 PM
Index 8
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Rasl2tp
Driver 48.11 KB (49,264 bytes)

Name [009] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Name WAN Miniport (PPTP)
Installed True
PNP Device ID ROOT\MS_PPTPMINIPORT\0000
Last Reset 2/22/2000 12:35:01 PM
Index 9
Service Name PptpMiniport

IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name PptpMiniport
Driver 44.58 KB (45,648 bytes)

Name [010] Direct Parallel
Adapter Type Direct Parallel
Product Name Direct Parallel
Installed True
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 2/22/2000 12:35:01 PM
Index 10
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Raspti
Driver 16.48 KB (16,880 bytes)

Name [011] WAN Miniport (IP)
Adapter Type WAN Miniport (IP)
Product Name WAN Miniport (IP)
Installed True
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 2/22/2000 12:35:01 PM
Index 11
Service Name NdisWan
IP Address 0.0.0.0
IP Subnet 0.0.0.0
Default IP Gateway
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name NdisWan
Driver 86.92 KB (89,008 bytes)

[Protocol]

Item Value
Name MSAFD Tcpip [TCP/IP]
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 16 bytes
MaximumMessageSize 0 bytes
MessageOriented False
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData True
SupportsFragmentation Not Available
SupportsGracefulClosing True
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD Tcpip [UDP/IP]
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 16 bytes
MaximumMessageSize 65467 bytes
MessageOriented True
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting True

Name RSVP UDP Service Provider
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 16 bytes
MaximumMessageSize 65467 bytes
MessageOriented True
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting True

Name RSVP TCP Service Provider
ConnectionlessService False

GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 16 bytes
MaximumMessageSize 0 bytes
MessageOriented False
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData True
SupportsFragmentation Not Available
SupportsGracefulClosing True
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{5F1FACD0-A684-4F4B-9C30-A67EABC47482}]SEQPACKET 0
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{5F1FACD0-A684-4F4B-9C30-A67EABC47482}]DATAGRAM 0
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{8C6819CD-C4A5-4DD5-9716-5D48BD7D127A}]SEQPACKET 1
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{8C6819CD-C4A5-4DD5-9716-5D48BD7D127A}]DATAGRAM 1
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{F4DF8431-3C64-49DE-8FAC-4E1A7CFFB040}]SEQPACKET 2
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False

SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{F4DF8431-3C64-49DE-8FAC-4E1A7CFFB040}]DATAGRAM 2
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{A94F4142-38F0-4F62-A834-3DA38C65F9EA}]SEQPACKET 3
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{A94F4142-38F0-4F62-A834-3DA38C65F9EA}]DATAGRAM 3
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{D810B7BF-105F-4D57-8BB4-989393ABD88D}]SEQPACKET 4
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{D810B7BF-105F-4D57-8BB4-989393ABD88D}]DATAGRAM 4
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{8FA7414C-964C-4AFF-9267-8201459FF2BB}]SEQPACKET 5
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False

SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{8FA7414C-964C-4AFF-9267-8201459FF2BB}]DATAGRAM 5
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{35698883-5357-4FEA-8782-BED2B3CA133A}]SEQPACKET 6
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{35698883-5357-4FEA-8782-BED2B3CA133A}]DATAGRAM 6
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{FD81683D-EB0D-4F78-B4FC-8A1488C772E3}]SEQPACKET 7
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{FD81683D-EB0D-4F78-B4FC-8A1488C772E3}]DATAGRAM 7
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{C1FBC3CE-BB75-4765-99E6-B13E6491CB15}]SEQPACKET 8
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False

SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{C1FBC3CE-BB75-4765-99E6-B13E6491CB15}]DATAGRAM 8
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsFragmentation Not Available
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

[WinSock]

Item Value
File c:\winnt\system32\winsock.dll
Version 3.10
Size 2.80 KB (2,864 bytes)

File c:\winnt\system32\wsock32.dll
Version 5.00.2120.1
Size 21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

Item Value
Name Communications Port(COM1)
Status OK
PNP Device ID ACPI\PNP0501\1
Maximum Input Buffer Size 0
Maximum Output Buffer Size False
Settable Baud Rate True
Settable Data Bits True
Settable Flow Control True
Settable Parity True
Settable Parity Check True
Settable Stop Bits True
Settable RLSD True
Supports RLSD True
Supports 16 Bit Mode False
Supports Special Characters False
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity NONE
Busy 0
Abort Read/Write on Error 0
Binary Mode Enabled -1
Continue XMit on XOff 0
CTS Outflow Control 0
Discard NULL Bytes 0
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type ENABLE
EOF Character 0
Error Replace Character 0
Error Replacement Enabled 0
Event Character 0
Parity Check Enabled 0
RTS Flow Control Type ENABLE
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Number 4
I/O Port 0x03F8-0x03FF
Driver 60.95 KB (62,416 bytes)

Name Communications Port(COM2)
Status OK
PNP Device ID ACPI\PNP0501\2
Maximum Input Buffer Size 0
Maximum Output Buffer Size False
Settable Baud Rate True
Settable Data Bits True
Settable Flow Control True
Settable Parity True
Settable Parity Check True
Settable Stop Bits True
Settable RLSD True
Supports RLSD True
Supports 16 Bit Mode False
Supports Special Characters False
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity NONE
Busy 0
Abort Read/Write on Error 0
Binary Mode Enabled -1

Continue XMit on XOff 0
CTS Outflow Control 0
Discard NULL Bytes 0
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type ENABLE
EOF Character 0
Error Replace Character 0
Error Replacement Enabled 0
Event Character 0
Parity Check Enabled 0
RTS Flow Control Type ENABLE
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Number 3
I/O Port 0x02F8-0x02FF
Driver 60.95 KB (62,416 bytes)

[Parallel]

Item Value
Name LPT1
PNP Device ID ACPI\PNP0400\1

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item Value
Drive A:
Description 3 1/2 Inch Floppy Drive

Drive C:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 4.19 GB (4,499,193,856 bytes)
Free Space 2.49 GB (2,675,982,336 bytes)
Volume Name
Volume Serial Number 448CD93F
Partition Disk #0, Partition #0
Partition Size 4.19 GB (4,499,195,904 bytes)
Starting Offset 32256 bytes
Drive Description Disk drive
Drive Manufacturer (Standard disk drives)
Drive Model IBM DCHS04Y SCSI Disk Device
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSIBus 0
Drive SCSILogicalUnit 0
Drive SCSIPort 3
Drive SCSITargetId 0
Drive SectorsPerTrack 63
Drive Size 4507453440 bytes
Drive TotalCylinders 548
Drive TotalSectors 8803620
Drive TotalTracks 139740
Drive TracksPerCylinder 255

[SCSI]

Item Value
Name Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
Caption Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
Driver aic78xx
Status OK
PNP Device ID PCI\VEN_9004&DEV_7895&SUBSYS_78959004&REV_04\3&267A616A&0&30
Device ID PCI\VEN_9004&DEV_7895&SUBSYS_78959004&REV_04\3&267A616A&0&30
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 16
I/O Port 0x2200-0x22FF
Driver 55.58 KB (56,912 bytes)

Name Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
Caption Adaptec AHA-2940U/UW Dual/AHA-394xAU/AUW/AUWD PCI SCSI Controller
Driver aic78xx
Status OK
PNP Device ID PCI\VEN_9004&DEV_7895&SUBSYS_78959004&REV_04\3&267A616A&0&31
Device ID PCI\VEN_9004&DEV_7895&SUBSYS_78959004&REV_04\3&267A616A&0&31
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 16
I/O Port 0x2300-0x23FF
Driver 55.58 KB (56,912 bytes)

[Printing]

Name Port Name Server Name
No printing information

[Problem Devices]

Device PNP Device ID Error Code
Not Available ACPI\IBM37C0\4&3D912C8&0 28

[USB]

Device PNP Device ID
USB Root Hub USB\ROOT_HUB\4&372644EA&0

[Software Environment]

[Following are sub-categories of this main category]

[Drivers]

Name Description File Type Started Start Mode State Status Error Control
Accept Pause Accept Stop
abiosdsk Abiosdsk Not Available Kernel Driver False Disabled Stopped OK
Ignore False False
abp480n5 abp480n5 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
acpi Microsoft ACPI Driver c:\winnt\system32\drivers\acpi.sys Kernel Driver True Boot
Start Running OK Normal False True
acpiec ACPIEC c:\winnt\system32\drivers\acpiec.sys Kernel Driver False Disabled Stopped
OK Normal False False
adpu160m adpu160m Not Available Kernel Driver False Disabled Stopped OK
Normal False False
afd AFD Networking Support Environment c:\winnt\system32\drivers\afd.sys Kernel Driver
True Auto Start Running OK Normal False True
aha154x Aha154x Not Available Kernel Driver False Disabled Stopped OK
Normal False False
aic116x aic116x Not Available Kernel Driver False Disabled Stopped OK
Normal False False
aic78u2 aic78u2 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
aic78xx aic78xx c:\winnt\system32\drivers\aic78xx.sys Kernel Driver True Boot Start Running
OK Normal False True
ami0nt ami0nt Not Available Kernel Driver False Disabled Stopped OK
Normal False False
amsint amsint Not Available Kernel Driver False Disabled Stopped OK
Normal False False
arp1394 1394 ARP Client Protocol c:\winnt\system32\drivers\arp1394.sys Kernel Driver False
Demand Start Stopped OK Normal False False
asc asc Not Available Kernel Driver False Disabled Stopped OK
Normal False False
asc3350p asc3350p Not Available Kernel Driver False Disabled Stopped OK
Normal False False
asc3550 asc3550 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
asyncmac RAS Asynchronous Media Driver c:\winnt\system32\drivers\asyncmac.sys Kernel Driver
False Demand Start Stopped OK Normal False False
atapi Standard IDE/ESDI Hard Disk Controller c:\winnt\system32\drivers\atapi.sys Kernel Driver
True Boot Start Running OK Normal False True
atdisk Atdisk Not Available Kernel Driver False Disabled Stopped OK
Ignore False False
atmarpc ATM ARP Client Protocol c:\winnt\system32\drivers\atmarpc.sys Kernel Driver False
Demand Start Stopped OK Normal False False
audstub Audio Stub Driver c:\winnt\system32\drivers\audstub.sys Kernel Driver True Demand
Start Running OK Normal False True
beep Beep c:\winnt\system32\drivers\beep.sys Kernel Driver True System Start Running
OK Normal False True
buslogic BusLogic Not Available Kernel Driver False Disabled Stopped OK
Normal False False
cd20xrnt cd20xrnt Not Available Kernel Driver False Disabled Stopped OK
Normal False False
cdaudio Cdaudio c:\winnt\system32\drivers\cdaudio.sys Kernel Driver False System Start Stopped
OK Ignore False False
cdfs Cdfs c:\winnt\system32\drivers\cdfs.sys File System Driver True Disabled Running
OK Normal False True
cdrom CD-ROM Driver c:\winnt\system32\drivers\cdrom.sys Kernel Driver True System
Start Running OK Normal False True
changer Changer Not Available Kernel Driver False System Start Stopped OK
Ignore False False
cpqarray Cpqarray Not Available Kernel Driver False Disabled Stopped OK
Normal False False
cpqfcalm cpqfcalm Not Available Kernel Driver False Disabled Stopped OK
Normal False False
cpqfws2e cpqfws2e Not Available Kernel Driver False Disabled Stopped OK
Normal False False
dac960nt dac960nt Not Available Kernel Driver False Disabled Stopped OK
Normal False False
deckzpsx deckzpsx Not Available Kernel Driver False Disabled Stopped OK
Normal False False
dfsdriver DfsDriver c:\winnt\system32\drivers\dfs.sys File System Driver True Boot Start Running
OK Normal False True
disk Disk Driver c:\winnt\system32\drivers\disk.sys Kernel Driver True Boot Start Running
OK Normal False True
diskperf Diskperf c:\winnt\system32\drivers\diskperf.sys Kernel Driver True Boot Start Running
OK Normal False True
dmboot dmboot c:\winnt\system32\drivers\dmboot.sys Kernel Driver False Disabled Stopped
OK Normal False False
dmio Logical Disk Manager Driver c:\winnt\system32\drivers\dmio.sys Kernel Driver True
Boot Start Running OK Normal False True
dmload dmload c:\winnt\system32\drivers\dmload.sys Kernel Driver True Boot Start Running
OK Normal False True
e100b Intel PRO Adapter Driver c:\winnt\system32\drivers\e100bnt5.sys Kernel Driver True
Demand Start Running OK Normal False True
efs EFS c:\winnt\system32\drivers\efs.sys File System Driver True Disabled Running
OK Normal False True
fastfat Fastfat c:\winnt\system32\drivers\fastfat.sys File System Driver True Disabled Running
OK Normal False True
fd16_700 Fd16_700 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
fdc Floppy Disk Controller Driver c:\winnt\system32\drivers\fdc.sys Kernel Driver True
Demand Start Running OK Normal False True
fireport fireport Not Available Kernel Driver False Disabled Stopped OK
Normal False False
flashpnt flashpnt Not Available Kernel Driver False Disabled Stopped OK
Normal False False
flpydisk Floppy Disk Driver c:\winnt\system32\drivers\flpydisk.sys Kernel Driver True Demand
Start Running OK Normal False True
ftdisk Volume Manager Driver c:\winnt\system32\drivers\ftdisk.sys Kernel Driver True Boot
Start Running OK Normal False True
gpc Generic Packet Classifier c:\winnt\system32\drivers\msgpc.sys Kernel Driver True Demand
Start Running OK Normal False True
i8042prt i8042 Keyboard and PS/2 Mouse Port Driver c:\winnt\system32\drivers\i8042prt.sys Kernel Driver
True System Start Running OK Normal False True

```
ibmfe IBM 10/100 Ethernet PCI Adapter NT Driver c:\winnt\system32\drivers\ibmfent5.sysKernel Driver
True Demand Start Running OK Normal False True
ini910u ini910u Not Available Kernel Driver False Disabled Stopped OK
Normal False False
intelide IntelIde Not Available Kernel Driver False Disabled Stopped OK
Normal False False
ipfilterdriverIP Traffic Filter Driver c:\winnt\system32\drivers\ipfltdrv.sys Kernel Driver False Demand
Start Stopped OK Normal False False
ipinip IP in IP Tunnel Driver c:\winnt\system32\drivers\ipinip.sys Kernel Driver False Demand
Start Stopped OK Normal False False
ipnat IP Network Address Translator c:\winnt\system32\drivers\ipnat.sys Kernel Driver False
Demand Start Stopped OK Normal False False
ipsec IPSEC driver c:\winnt\system32\drivers\ipsec.sys Kernel Driver True Demand
Start Running OK Normal False True
ipsraidn ipsraidn Not Available Kernel Driver False Disabled Stopped OK
Normal False False
isapnp PnP ISA/EISA Bus Driver c:\winnt\system32\drivers\isapnp.sys Kernel Driver True
Boot Start Running OK Critical False True
kbdclass Keyboard Class Driver c:\winnt\system32\drivers\kbdclass.sysKernel Driver True System
Start Running OK Normal False True
ksecdd KSecDD c:\winnt\system32\drivers\ksecdd.sys Kernel Driver True Boot Start Running
OK Normal False True
lbrtfdc lbrtfdc Not Available Kernel Driver False System StartStopped OK
Ignore False False
lp6nds35 lp6nds35 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
mnmdd mnmdd c:\winnt\system32\drivers\mnmdd.sys Kernel Driver True System Start Running
OK Ignore False True
modem Modem c:\winnt\system32\drivers\modem.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
mouclass Mouse Class Driver c:\winnt\system32\drivers\mouclass.sys Kernel Driver True
System Start Running OK Normal False True
mountmgr MountMgr c:\winnt\system32\drivers\mountmgr.sys Kernel Driver True Boot
Start Running OK Normal False True
mraid35x mraid35x Not Available Kernel Driver False Disabled Stopped OK
Normal False False
mrxsmb MRXSMB c:\winnt\system32\drivers\mrxsmb.sysFile System Driver True System Start Running
OK Normal False True
msfs Msfs c:\winnt\system32\drivers\msfs.sys File System Driver True System Start Running
OK Normal False True
mskssrv Microsoft Streaming Service Proxy c:\winnt\system32\drivers\mskssrv.sysKernel Driver False
Demand Start Stopped OK Normal False False
mspclock Microsoft Streaming Clock Proxy c:\winnt\system32\drivers\mspclock.sys Kernel Driver
False Demand Start Stopped OK Normal False False
mspqm Microsoft Streaming Quality Manager Proxy c:\winnt\system32\drivers\mspqm.sys Kernel Driver
False Demand Start Stopped OK Normal False False
mup Mup c:\winnt\system32\drivers\mup.sys File System Driver True Boot Start Running
OK Normal False True
ncrc710 Ncrc710 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
ndis NDIS System Driver c:\winnt\system32\drivers\ndis.sys Kernel Driver True Boot
Start Running OK Normal False True
ndistapi Remote Access NDIS TAPI Driver c:\winnt\system32\drivers\ndistapi.sys Kernel Driver True
Demand Start Running OK Normal False True
ndiswan Remote Access NDIS WAN Driver c:\winnt\system32\drivers\ndiswan.sysKernel Driver True
Demand Start Running OK Normal False True
ndproxy NDIS Proxy c:\winnt\system32\drivers\ndproxy.sysKernel Driver True Demand Start
Running OK Normal False True
netbios NetBIOS Interface c:\winnt\system32\drivers\netbios.sys File System Driver True System
Start Running OK Normal False True
netbt NetBios over Tcpip c:\winnt\system32\drivers\netbt.sys Kernel Driver True System
Start Running OK Normal False True
netdetect NetDetect c:\winnt\system32\drivers\netdtect.sys Kernel Driver False Demand Start
Stopped OK Normal False False
npfs Npfs c:\winnt\system32\drivers\npfs.sys File System Driver True System StartRunning
OK Normal False True
ntfs Ntfs c:\winnt\system32\drivers\ntfs.sys File System Driver True Disabled Running
OK Normal False True
null Null c:\winnt\system32\drivers\null.sys Kernel Driver True System StartRunning
OK Normal False True
nwlnkflt IPX Traffic Filter Driver c:\winnt\system32\drivers\nwlnkflt.sysKernel Driver False Demand
Start Stopped OK Normal False False
nwlnkfwd IPX Traffic Forwarder Driver c:\winnt\system32\drivers\nwlnkfwd.sys Kernel Driver
False Demand Start Stopped OK Normal False False
openhci Microsoft USB Open Host Controller Driver c:\winnt\system32\drivers\openhci.sys Kernel Driver
True Demand Start Running OK Normal False True
parallel Parallel class driver c:\winnt\system32\drivers\parallel.sys Kernel Driver True Demand
Start Running OK Normal False True
parport Parallel port driver c:\winnt\system32\drivers\parport.sys Kernel Driver True System
Start Running OK Ignore False True
partmgr PartMgr c:\winnt\system32\drivers\partmgr.sys Kernel Driver True Boot Start Running
OK Normal False True
parvdm ParVdm c:\winnt\system32\drivers\parvdm.sys Kernel Driver True Auto Start Running
OK Ignore False True
pci PCI Bus Driver c:\winnt\system32\drivers\pci.sys Kernel Driver True Boot
Start Running OK Critical False True
pcidump PCIDump Not Available Kernel Driver False System StartStopped OK
Ignore False False
pciide PCIIde c:\winnt\system32\drivers\pciide.sys Kernel Driver True Boot Start Running
OK Normal False True
pcmcia Pcmcia c:\winnt\system32\drivers\pcmcia.sys Kernel Driver False Disabled Stopped
OK Normal False False
pcnet AMD PCNET Compatable Adapter Driver c:\winnt\system32\drivers\pcntn5m.sysKernel Driver
True Demand Start Running OK Normal False True
pdcomp PDCOMP Not Available Kernel Driver False Demand Start Stopped
OK Ignore False False
pdframe PDFRAME Not Available Kernel Driver False Demand Start Stopped
OK Ignore False False
pdreli PDRELI Not Available Kernel Driver False Demand Start Stopped
OK Ignore False False
pdrframe PDRFRAME Not Available Kernel Driver False Demand Start
Stopped OK Ignore False False
pptpminiportWAN Miniport (PPTP) c:\winnt\system32\drivers\raspptp.sys Kernel Driver True Demand
Start Running OK Normal False True
ptilink Direct Parallel Link Driver c:\winnt\system32\drivers\ptilink.sys Kernel Driver True
Demand Start Running OK Normal False True
ql1080 ql1080 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
ql10wnt Ql10wnt Not Available Kernel Driver False Disabled Stopped OK
Normal False False
ql1240 ql1240 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
```

```
ql2100 ql2100 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
rasacd Remote Access Auto Connection Driver c:\winnt\system32\drivers\rasacd.sys Kernel Driver
True System Start Running OK Normal False True
rasl2tp WAN Miniport (L2TP) c:\winnt\system32\drivers\rasl2tp.sys Kernel Driver True Demand
Start Running OK Normal False True
raspti Direct Parallel c:\winnt\system32\drivers\raspti.sys Kernel Driver True Demand
Start Running OK Normal False True
rca Microsoft Streaming Network Raw Channel Access c:\winnt\system32\drivers\rca.sys Kernel Driver
False Demand Start Stopped OK Normal False False
rdbss Rdbss c:\winnt\system32\drivers\rdbss.sys File System Driver True System Start Running
OK Normal False True
rdpwd RDPWD c:\winnt\system32\drivers\rdpwd.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
redbook Digital CD Audio Playback Filter Driver c:\winnt\system32\drivers\redbook.sys Kernel Driver
False System Start Stopped OK Normal False False
s3m s3m c:\winnt\system32\drivers\s3m.sys Kernel Driver True Demand Start
Running OK Ignore False True
serenum Serenum Filter Driver c:\winnt\system32\drivers\serenum.sys Kernel Driver True Demand
Start Running OK Normal False True
serial Serial port driver c:\winnt\system32\drivers\serial.sys Kernel Driver True System
Start Running OK Ignore False True
sfloppy Sfloppy c:\winnt\system32\drivers\sfloppy.sys Kernel Driver False System Start Stopped
OK Ignore False False
sglfb sglfb Not Available Kernel Driver False System Start Stopped OK
Normal False False
simbad Simbad Not Available Kernel Driver False Disabled Stopped OK
Normal False False
sparrow Sparrow Not Available Kernel Driver False Disabled Stopped OK
Normal False False
spud Special Purpose Utility Driver c:\winnt\system32\drivers\spud.sys Kernel Driver True
Demand Start Running OK Normal False True
srv Srv c:\winnt\system32\drivers\srv.sys File System Driver True Demand Start
Running OK Normal False True
swenum Software Bus Driver c:\winnt\system32\drivers\swenum.sys Kernel Driver True Demand
Start Running OK Normal False True
symc810 symc810 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
symc8xx symc8xx Not Available Kernel Driver False Disabled Stopped OK
Normal False False
sym_hi sym_hi Not Available Kernel Driver False Disabled Stopped OK
Normal False False
tcpip TCP/IP Protocol Driver c:\winnt\system32\drivers\tcpip.sys Kernel Driver True System
Start Running OK Normal False True
tdasync TDASYNC c:\winnt\system32\drivers\tdasync.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
tdipx TDIPX c:\winnt\system32\drivers\tdipx.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
tdnetb TDNETB c:\winnt\system32\drivers\tdnetb.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
tdpipe TDPIPE c:\winnt\system32\drivers\tdpipe.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
tdspx TDSPX c:\winnt\system32\drivers\tdspx.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
tdtcp TDTCP c:\winnt\system32\drivers\tdtcp.sys Kernel Driver False Demand Start
Stopped OK Ignore False False
termdd Terminal Device Driver c:\winnt\system32\drivers\termdd.sys Kernel Driver False
Disabled Stopped OK Normal False False
tga tga Not Available Kernel Driver False System Start Stopped OK
Ignore False False
udfs Udfs c:\winnt\system32\drivers\udfs.sys File System Driver False Disabled Stopped
OK Normal False False
ultra66 ultra66 Not Available Kernel Driver False Disabled Stopped OK
Normal False False
update Microcode Update Driver c:\winnt\system32\drivers\update.sys Kernel Driver True Demand
Start Running OK Normal False True
usbhub Microsoft USB Standard Hub Driver c:\winnt\system32\drivers\usbhub.sys Kernel Driver True
Demand Start Running OK Normal False True
vgasave VgaSave c:\winnt\system32\drivers\vga.sys Kernel Driver True System Start Running
OK Ignore False True
wanarp Remote Access IP ARP Driver c:\winnt\system32\drivers\wanarp.sys Kernel Driver True
Demand Start Running OK Normal False True
wdica WDICA Not Available Kernel Driver False Demand Start Stopped
OK Ignore False False

[Environment Variables]

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
Os2LibPath %SystemRoot%\system32\os2\dll; <SYSTEM>
Path %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\MSSQL7\BINN;
<SYSTEM>
windir %SystemRoot% <SYSTEM>
OS Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_LEVEL 6 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 6 Model 7 Stepping 3, GenuineIntel <SYSTEM>
PROCESSOR_REVISION 0703 <SYSTEM>
NUMBER_OF_PROCESSORS 1 <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp SQLCLIENT40\Administrator
TMP %USERPROFILE%\Local Settings\Temp SQLCLIENT40\Administrator

[Jobs]

[ Following are sub-categories of this main category ]

[Print]

Document Size Owner Notify Status Time Submitted Start Time Until Time Elapsed
Time Pages Printed Job ID Priority Parameters Driver Name Print Processor
Host Print Queue Data Type Name
No print jobs


[Network Connections]

Local Name Remote Name Type Status User Name
No network connections information
```

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Max Working Set	Start Time	Version	Size	File Date
system idle process	Not Available	0	0	Not Available	Not Available	Not Available	Unknown	Unknown	Unknown
system	Not Available	8	8	0	1413120	Not Available	Unknown	Unknown	Unknown
smss.exe	c:\winnt\system32\smss.exe	156	11	204800	1413120	2/22/2000 5:35:26 PM	5.00.2090.1	42.77 KB (43,792 bytes)	Not Available
csrss.exe	Not Available	184	13	Not Available	Not Available	2/22/2000 5:35:38 PM	Unknown	Unknown	Unknown
winlogon.exe	c:\winnt\system32\winlogon.exe	180	13	204800	1413120	2/22/2000 5:35:41 PM	5.00.2116.1	171.27 KB (175,376 bytes)	Not Available
services.exe	c:\winnt\system32\services.exe	232	9	204800	1413120	2/22/2000 5:35:47 PM	5.00.2106.1	87.27 KB (89,360 bytes)	Not Available
lsass.exe	c:\winnt\system32\lsass.exe	244	13	204800	1413120	2/22/2000 5:35:47 PM	5.00.2121.1	32.77 KB (33,552 bytes)	Not Available
svchost.exe	c:\winnt\system32\svchost.exe	420	8	204800	1413120	2/22/2000 5:36:08 PM	5.00.2090.1	7.77 KB (7,952 bytes)	Not Available
spoolsv.exe	c:\winnt\system32\spoolsv.exe	452	8	204800	1413120	2/22/2000 5:36:12 PM	5.00.2107.1	43.77 KB (44,816 bytes)	Not Available
msdtc.exe	c:\winnt\system32\msdtc.exe	480	8	204800	1413120	2/22/2000 5:36:12 PM	1999.8.3413.3	6.77 KB (6,928 bytes)	Not Available
svchost.exe	c:\winnt\system32\svchost.exe	588	8	204800	1413120	2/22/2000 5:36:23 PM	5.00.2090.1	7.77 KB (7,952 bytes)	Not Available
llssrv.exe	c:\winnt\system32\llssrv.exe	612	9	204800	1413120	2/22/2000 5:36:23 PM	5.00.2090.1	113.77 KB (116,496 bytes)	Not Available
regsvc.exe	c:\winnt\system32\regsvc.exe	676	8	204800	1413120	2/22/2000 5:36:24 PM	5.00.2091.1	63.77 KB (65,296 bytes)	Not Available
mstask.exe	c:\winnt\system32\mstask.exe	740	8	204800	1413120	2/22/2000 5:36:27 PM	4.71.2113.1	114.77 KB (117,520 bytes)	Not Available
tcpsvcs.exe	c:\winnt\system32\tcpsvcs.exe	844	8	204800	1413120	2/22/2000 5:36:47 PM	5.00.2090.1	24.77 KB (25,360 bytes)	Not Available
inetinfo.exe	c:\winnt\system32\inetsrv\inetinfo.exe	880	8	204800	1413120	2/22/2000 5:36:48 PM	5.00.0984	14.27 KB (14,608 bytes)	Not Available
dfssvc.exe	c:\winnt\system32\dfssvc.exe	924	8	204800	1413120	2/22/2000 5:36:50 PM	5.00.2124.1	95.77 KB (98,064 bytes)	Not Available
explorer.exe	c:\winnt\explorer.exe	732	8	204800	1413120	2/23/2000 10:34:23 AM	5.00.2919.3800	232.77 KB (238,352 bytes)	Not Available
dllhost.exe	Not Available	320	8	Not Available	Not Available	2/23/2000 10:36:18 AM	Unknown	Unknown	Unknown
mmc.exe	c:\winnt\system32\mmc.exe	872	8	204800	1413120	2/23/2000 11:11:36 AM	5.00.2115.1	589.27 KB (603,408 bytes)	Not Available
winmgmt.exe	c:\winnt\system32\wbem\winmgmt.exe	912	8	204800	1413120	2/23/2000 11:13:01 AM	1.50.1025.0015	164.05 KB (167,991 bytes)	Not Available
rsvp.exe	c:\winnt\system32\rsvp.exe	2084	8	204800	1413120	2/23/2000 11:15:31 AM	5.00.2120.1	170.77 KB (174,864 bytes)	Not Available
mdm.exe	c:\winnt\system32\mdm.exe	2040	8	204800	1413120	2/23/2000 11:16:11 AM	6.00.8424	121.29 KB (124,200 bytes)	Not Available

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
mdm.exe	6.00.8424	121.29 KB (124,200 bytes)	2/8/2000 10:19:06 AM	Microsoft Corporation	c:\winnt\system32\mdm.exe
traffic.dll	5.00.2090.1	30.77 KB (31,504 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\traffic.dll
rsvp.exe	5.00.2120.1	170.77 KB (174,864 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\rsvp.exe
perfos.dll	5.0	21.27 KB (21,776 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\perfos.dll
ieinfo5.ocx	5.00.2107.1	96.27 KB (98,576 bytes)	2/8/2000 3:25:01 PM	Microsoft Corporation	c:\progra~1\common~1\micros~1\msinfo\ieinfo5.ocx
rapilib.dll	5.00.2120.1	24.77 KB (25,360 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\rapilib.dll
rsvpsp.dll	5.00.2120.1	74.77 KB (76,560 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\rsvpsp.dll
ntevt.dll	1.50.1025.0002	176.06 KB (180,290 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\ntevt.dll
ntmarta.dll	5.00.2119.1	98.27 KB (100,624 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\ntmarta.dll
psapi.dll	5.00.2090.1	28.27 KB (28,944 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\psapi.dll
framedyn.dll	1.50.1025.0002	164.05 KB (167,988 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll	1.50.1025.0016	1.02 MB (1,065,018 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\cimwin32.dll
adsldp.dll	5.00.2118.1	119.27 KB (122,128 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\adsldp.dll
provthrd.dll	1.50.1025.0001	68.08 KB (69,713 bytes)	2/8/2000 3:25:22 PM	Microsoft Corporation	c:\winnt\system32\wbem\provthrd.dll
dsprov.dll	1.50.1025.0002	192.06 KB (196,670 bytes)	2/8/2000 3:25:23 PM	Microsoft Corporation	c:\winnt\system32\wbem\dsprov.dll
mofd.dll	1.50.1025.0009	128.05 KB (131,120 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\mofd.dll
wmiprov.dll	1.50.1025.0002	88.06 KB (90,177 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\wmiprov.dll
wbemess.dll	1.50.1025.0009	324.05 KB (331,827 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\wbemess.dll
wbemcore.dll	1.50.1025.0012	592.05 KB (606,260 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\wbemcore.dll
winmgmt.exe	1.50.1025.0015	164.05 KB (167,991 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\winmgmt.exe
fastprox.dll	1.50.1025.0009	144.08 KB (147,536 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\fastprox.dll
wbemsvc.dll	1.50.1025.0009	136.07 KB (139,339 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\wbemsvc.dll
wbemcomn.dll	1.50.1025.0009	688.05 KB (704,564 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\wbemcomn.dll
wbemprox.dll	1.50.1025.0009	40.05 KB (41,012 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\wbem\wbemprox.dll
rassapi.dll	5.00.2116.1	14.27 KB (14,608 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\rassapi.dll
adsnt.dll	5.00.2118.1	193.77 KB (198,416 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\adsnt.dll
dbghelp.dll	5.00.2128.1	77.27 KB (79,120 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\dbghelp.dll
localsec.dll	5.00.2099.1	226.77 KB (232,208 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\localsec.dll
devmgr.dll	5.00.2109.1	215.27 KB (220,432 bytes)	9/9/1999 8:00:00 PM	Microsoft Corporation	c:\winnt\system32\devmgr.dll

filemgmt.dll 5.00.2116.1 287.27 KB (294,160 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\filemgmt.dll
pdh.dll 5.00.1838.1 135.77 KB (139,024 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\pdh.dll
smlogcfg.dll 5.00.2107.1 272.77 KB (279,312 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\smlogcfg.dll
cabinet.dll 5.00.2090.1 54.77 KB (56,080 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\cabinet.dll
msinfo32.dll 5.00.2121.1 306.27 KB (313,616 bytes) 2/8/2000 3:25:33 PM Microsoft Corporation c:\program files\common files\microsoft shared\msinfo\msinfo32.dll
riched20.dll 5.30.20.1200 419.77 KB (429,840 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\riched20.dll
riched32.dll 5.00.2090.1 3.77 KB (3,856 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\riched32.dll
els.dll 5.00.2108.1 146.77 KB (150,288 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\els.dll
ntmsmgr.dll 1,0,0,1 427.27 KB (437,520 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation and HighGround Systems, Inc. c:\winnt\system32\ntmsmgr.dll
mmfutil.dll 1.50.1025.0005 32.06 KB (32,834 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mmfutil.dll
logdrive.dll 1.50.1025.0004 200.07 KB (204,868 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\logdrive.dll
dfrgres.dll 5.00.2109.1 27.50 KB (28,160 bytes) 9/9/1999 8:00:00 PM Executive Software International, Inc. c:\winnt\system32\dfrgres.dll
dfrgsnap.dll 5.00.2109.1 41.77 KB (42,768 bytes) 9/9/1999 8:00:00 PM Executive Software International, Inc. c:\winnt\system32\dfrgsnap.dll
dmdskres.dll 2121.1.286.1 119.00 KB (121,856 bytes) 9/9/1999 8:00:00 PM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskres.dll
dmutil.dll 2121.1.286.1 41.77 KB (42,768 bytes) 9/9/1999 8:00:00 PM VERITAS Software Corp. c:\winnt\system32\dmutil.dll
ntmsapi.dll 5.00.1948.1 53.27 KB (54,544 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\ntmsapi.dll
dmdskmgr.dll 2121.1.286.1 158.27 KB (162,064 bytes) 9/9/1999 8:00:00 PM Microsoft Corp., VERITAS Software c:\winnt\system32\dmdskmgr.dll
mycomput.dll 5.00.2090.1 107.77 KB (110,352 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mycomput.dll
mmcndmgr.dll 5.00.2108.1 815.27 KB (834,832 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mmcndmgr.dll
mmc.exe 5.00.2115.1 589.27 KB (603,408 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mmc.exe
usp10.dll 1.0306.2115.1 307.27 KB (314,640 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\usp10.dll
thumbvw.dll 5.00.2919.3800 183.27 KB (187,664 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\thumbvw.dll
mshtmled.dll 5.00.2919.3800 230.27 KB (235,792 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mshtmled.dll
webvw.dll 5.00.2919.3800 1.06 MB (1,113,360 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\webvw.dll
msls31.dll 3.10.337.0 144.77 KB (148,240 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msls31.dll
msdbg.dll 6.00.8424 67.50 KB (69,120 bytes) 2/8/2000 10:19:06 AM Microsoft Corporation c:\winnt\system32\msdbg.dll
pdm.dll 6.00.8424 179.27 KB (183,574 bytes) 2/8/2000 10:19:06 AM Microsoft Corporation c:\winnt\system32\pdm.dll
mshtml.dll 5.00.2919.3800 2.25 MB (2,355,984 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mshtml.dll
mlang.dll 5.00.2919.3800 509.77 KB (522,000 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mlang.dll
urlmon.dll 5.00.2919.3800 426.77 KB (437,008 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\urlmon.dll
browselc.dll 5.00.2919.3800 34.50 KB (35,328 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\browselc.dll
shdoclc.dll 5.00.2919.3800 324.50 KB (332,288 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\shdoclc.dll
wininet.dll 5.00.2919.3800 456.77 KB (467,728 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\wininet.dll
linkinfo.dll 5.00.2091.1 15.77 KB (16,144 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\linkinfo.dll
hhsetup.dll 4.74.8576 66.27 KB (67,856 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\hhsetup.dll
mmcshext.dll 5.00.2108.1 24.27 KB (24,848 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mmcshext.dll
faxshell.dll 5.00.2101.1 8.27 KB (8,464 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\faxshell.dll
msacm32.dll 5.00.2113.1 65.27 KB (66,832 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msacm32.dll
avifil32.dll 5.00.2113.1 76.27 KB (78,096 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\avifil32.dll
msvfw32.dll 5.00.2113.1 113.77 KB (116,496 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msvfw32.dll
docprop2.dll 5.00.2115.1 297.77 KB (304,912 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\docprop2.dll
powrprof.dll 5.00.2919.3800 13.27 KB (13,584 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\powrprof.dll
batmeter.dll 5.00.2919.3800 20.27 KB (20,752 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\batmeter.dll
stobject.dll 5.00.2120.1 81.27 KB (83,216 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\stobject.dll
webcheck.dll 5.00.2919.3800 251.77 KB (257,808 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\webcheck.dll
netui1.dll 5.00.2107.1 209.77 KB (214,800 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\netui1.dll
netui0.dll 5.00.2107.1 70.27 KB (71,952 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\netui0.dll
ntlanman.dll 5.00.2109.1 35.27 KB (36,112 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\ntlanman.dll
ntshrui.dll 5.00.2090.1 46.77 KB (47,888 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\ntshrui.dll
mydocs.dll 5.00.2919.3800 55.77 KB (57,104 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mydocs.dll
browseui.dll 5.00.2919.3800 791.77 KB (810,768 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\browseui.dll
shdocvw.dll 5.00.2919.3800 1.05 MB (1,103,632 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\shdocvw.dll
explorer.exe 5.00.2919.3800 232.77 KB (238,352 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\explorer.exe
dfssvc.exe 5.00.2124.1 95.77 KB (98,064 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\dfssvc.exe
tpcc_com_ps.dll Not Available 5.00 KB (5,120 bytes) 1/28/2000 2:38:18 PM Not Available c:\inetpub\wwwroot\tpcc_com_ps.dll
tpcc_com_all.dll 1, 0, 0, 1 96.00 KB (98,304 bytes) 1/28/2000 2:38:19 PM c:\inetpub\wwwroot\tpcc_com_all.dll

dbmssocn.dll 1999.05.27 28.27 KB (28,944 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\dbmssocn.dll
ntwdblib.dll 1998.11.13 268.27 KB (274,704 bytes) 2/8/2000 4:14:21 PM Microsoft Corporation c:\winnt\system32\ntwdblib.dll
tpcc_dblib.dll Not Available 19.50 KB (19,968 bytes) 1/28/2000 2:38:17 PM Not Available c:\inetpub\wwwroot\tpcc_dblib.dll
tpcc_com.dllNot Available 11.50 KB (11,776 bytes) 1/28/2000 2:38:18 PM Not Available c:\inetpub\wwwroot\tpcc_com.dll
tpcc.dll 0, 4, 0, 0 78.00 KB (79,872 bytes) 1/28/2000 2:38:17 PM Microsoft c:\inetpub\wwwroot\tpcc.dll
mfc42.dll 6.00.8576.0 972.05 KB (995,383 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mfc42.dll
wam.dll 5.00.0984 69.77 KB (71,440 bytes) 2/8/2000 10:16:57 AM Microsoft Corporation c:\winnt\system32\inetsrv\wam.dll
odbcint.dll 3.520.4403.2 88.00 KB (90,112 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\odbcint.dll
comdlg32.dll 5.00.2919.3800 235.27 KB (240,912 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\comdlg32.dll
odbc32.dll 3.520.4403.2 212.27 KB (217,360 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\odbc32.dll
comsvcs.dll 1999.8.3413.5 1.17 MB (1,224,976 bytes) 2/8/2000 10:15:53 AM Microsoft Corporation c:\winnt\system32\comsvcs.dll
iislog.dll 5.00.0984 75.77 KB (77,584 bytes) 2/8/2000 10:16:48 AM Microsoft Corporation c:\winnt\system32\inetsrv\iislog.dll
ntlsapi.dll 5.00.2090.1 6.77 KB (6,928 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\ntlsapi.dll
wshnetbs.dll 5.00.2090.1 7.77 KB (7,952 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\wshnetbs.dll
httpext.dll 0.9.3938.5 415.77 KB (425,744 bytes) 2/8/2000 10:16:48 AM Microsoft Corporation c:\winnt\system32\inetsrv\httpext.dll
rpcproxy.dll 5.00.2128.1 16.27 KB (16,656 bytes) 2/8/2000 10:15:52 AM Microsoft Corporation c:\winnt\system32\rpcproxy\rpcproxy.dll
fpexedll.dll 4.0.2.3228 20.06 KB (20,541 bytes) 2/8/2000 10:19:00 AM Microsoft Corporation c:\program files\common files\microsoft shared\web server extensions\40\bin\fpexedll.dll
md5filt.dll 5.00.0984 32.77 KB (33,552 bytes) 2/8/2000 10:16:55 AM Microsoft Corporation c:\winnt\system32\inetsrv\md5filt.dll
gzip.dll 5.00.0984 30.27 KB (30,992 bytes) 2/8/2000 10:16:54 AM Microsoft Corporation c:\winnt\system32\inetsrv\gzip.dll
compfilt.dll 5.00.0984 22.27 KB (22,800 bytes) 2/8/2000 10:16:54 AM Microsoft Corporation c:\winnt\system32\inetsrv\compfilt.dll
sspifilt.dll 5.00.0984 42.77 KB (43,792 bytes) 2/8/2000 10:16:56 AM Microsoft Corporation c:\winnt\system32\inetsrv\sspifilt.dll
iscomlog.dll 5.00.0984 24.27 KB (24,848 bytes) 2/8/2000 10:16:49 AM Microsoft Corporation c:\winnt\system32\inetsrv\iscomlog.dll
lonsint.dll 5.00.0984 11.77 KB (12,048 bytes) 2/8/2000 10:16:49 AM Microsoft Corporation c:\winnt\system32\inetsrv\lonsint.dll
inetsloc.dll 5.00.0984 20.27 KB (20,752 bytes) 2/8/2000 10:16:50 AM Microsoft Corporation c:\winnt\system32\inetsloc.dll
iisfecnv.dll 5.00.0984 7.27 KB (7,440 bytes) 2/8/2000 10:16:48 AM Microsoft Corporation c:\winnt\system32\inetsrv\iisfecnv.dll
isatq.dll 5.00.0984 60.27 KB (61,712 bytes) 2/8/2000 10:16:51 AM Microsoft Corporation c:\winnt\system32\inetsrv\isatq.dll
infocomm.dll 5.00.0984 230.27 KB (235,792 bytes) 2/8/2000 10:16:49 AM Microsoft Corporation c:\winnt\system32\inetsrv\infocomm.dll
w3svc.dll 5.00.0984 345.27 KB (353,552 bytes) 2/8/2000 10:16:56 AM Microsoft Corporation c:\winnt\system32\inetsrv\w3svc.dll
security.dll 5.00.2112.1 5.77 KB (5,904 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\security.dll
svcext.dll 5.00.0984 39.27 KB (40,208 bytes) 2/8/2000 10:16:49 AM Microsoft Corporation c:\winnt\system32\inetsrv\svcext.dll
admexs.dll 5.00.0984 27.77 KB (28,432 bytes) 2/8/2000 10:16:48 AM Microsoft Corporation c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll 5.00.0984 44.77 KB (45,840 bytes) 2/8/2000 10:16:57 AM Microsoft Corporation c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll 5.00.0984 68.27 KB (69,904 bytes) 2/8/2000 10:16:49 AM Microsoft Corporation c:\winnt\system32\inetsrv\metadata.dll
iismap.dll 5.00.0984 56.27 KB (57,616 bytes) 2/8/2000 10:16:50 AM Microsoft Corporation c:\winnt\system32\iismap.dll
nsepm.dll 5.00.0984 43.27 KB (44,304 bytes) 2/8/2000 10:16:49 AM Microsoft Corporation c:\winnt\system32\inetsrv\nsepm.dll
admwprox.dll 5.00.0984 31.27 KB (32,016 bytes) 2/8/2000 10:16:50 AM Microsoft Corporation c:\winnt\system32\admwprox.dll
coadmin.dll 5.00.0984 39.27 KB (40,208 bytes) 2/8/2000 10:16:51 AM Microsoft Corporation c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll 5.00.0984 14.77 KB (15,120 bytes) 2/8/2000 10:16:48 AM Microsoft Corporation c:\winnt\system32\inetsrv\iisadmin.dll
rpcref.dll 5.00.0984 4.27 KB (4,368 bytes) 2/8/2000 10:16:49 AM Microsoft Corporation c:\winnt\system32\inetsrv\rpcref.dll
iisrtl.dll 5.00.0984 119.27 KB (122,128 bytes) 2/8/2000 10:16:50 AM Microsoft Corporation c:\winnt\system32\iisrtl.dll
inetinfo.exe 5.00.0984 14.27 KB (14,608 bytes) 2/8/2000 10:16:48 AM Microsoft Corporation c:\winnt\system32\inetsrv\inetinfo.exe
simptcp.dll 5.00.2106.1 19.27 KB (19,728 bytes) 2/8/2000 10:15:52 AM Microsoft Corporation c:\winnt\system32\simptcp.dll
tcpsvcs.exe 5.00.2090.1 24.77 KB (25,360 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\tcpsvcs.exe
msidle.dll 5.00.2919.3800 6.27 KB (6,416 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msidle.dll
mstask.exe 4.71.2113.1 114.77 KB (117,520 bytes) 2/8/2000 3:25:23 PM Microsoft Corporation c:\winnt\system32\mstask.exe
regsvc.exe 5.00.2091.1 63.77 KB (65,296 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\regsvc.exe
llsrpc.dll 5.00.2107.1 45.77 KB (46,864 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\llsrpc.dll
llssrv.exe 5.00.2090.1 113.77 KB (116,496 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\llssrv.exe
wmi.dll 5.00.2112.1 6.27 KB (6,416 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\wmi.dll
netshell.dll 5.00.2120.1 453.77 KB (464,656 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\netshell.dll
netman.dll 5.00.2120.1 88.77 KB (90,896 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\netman.dll
ntmsdba.dll 5.00.2108.1 167.27 KB (171,280 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\ntmsdba.dll
sens.dll 5.00.2090.1 35.77 KB (36,624 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\sens.dll
iashlpr.dll 5.00.2090.1 31.27 KB (32,016 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iashlpr.dll
iasacct.dll 5.00.2095.1 28.27 KB (28,944 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iasacct.dll
iasuserr.dll 5.00.2090.1 25.77 KB (26,384 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iasuserr.dll

iasnap.dll 5.00.2090.1 58.77 KB (60,176 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iasnap.dll
iaspipe.dll 5.00.2090.1 41.77 KB (42,768 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iaspipe.dll
expsrv.dll 6.0.8540 370.27 KB (379,152 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\expsrv.dll
vbajet32.dll 6.1.8268 30.27 KB (30,992 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\vbajet32.dll
msjtes40.dll 4.00.2927.6 232.27 KB (237,840 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msjtes40.dll
oledb32r.dll 2.50.4403.2 64.27 KB (65,808 bytes) 2/8/2000 3:25:08 PM Microsoft Corporation c:\program files\common files\system\ole db\oledb32r.dll
msdart32.dll 2.50.4403.0 24.27 KB (24,848 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msdart32.dll
oledb32.dll 2.50.4403.3 472.27 KB (483,600 bytes) 2/8/2000 3:25:08 PM Microsoft Corporation c:\program files\common files\system\ole db\oledb32.dll
msjint40.dll 4.00.2927.2 148.27 KB (151,824 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msjint40.dll
msjter40.dll 4.00.2927.2 52.27 KB (53,520 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msjter40.dll
mswstr10.dll4.00.2927.2 600.27 KB (614,672 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mswstr10.dll
msjet40.dll 4.00.2927.4 1.43 MB (1,495,312 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msjet40.dll
msjetoledb40.dll 4.00.2927.2 340.27 KB (348,432 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msjetoledb40.dll
iasrad.dll 5.00.2090.1 94.27 KB (96,528 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iasrad.dll
iassam.dll 5.00.2090.1 96.27 KB (98,576 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iassam.dll
iasads.dll 5.00.2112.1 73.77 KB (75,536 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iasads.dll
iaspolcy.dll 5.00.2090.1 25.27 KB (25,872 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iaspolcy.dll
iassvcs.dll 5.00.2090.1 58.27 KB (59,664 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iassvcs.dll
iassdo.dll 5.00.2090.1 262.27 KB (268,560 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iassdo.dll
ntmssvc.dll 5.00.2108.1 390.27 KB (399,632 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\ntmssvc.dll
ias.dll 5.00.2090.1 7.27 KB (7,440 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\ias.dll
es.dll 1999.8.3413.3 220.77 KB (226,064 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\es.dll
mtxoci.dll 1999.8.3413.3 101.77 KB (104,208 bytes) 2/8/2000 10:16:03 AM Microsoft Corporation c:\winnt\system32\mtxoci.dll
resutils.dll 5.00.2123.1 39.77 KB (40,720 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\resutils.dll
clusapi.dll 5.00.2123.1 49.27 KB (50,448 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\clusapi.dll
msvcp50.dll 5.00.7051 552.50 KB (565,760 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msvcp50.dll
xolehlp.dll 1999.8.3413.3 18.27 KB (18,704 bytes) 2/8/2000 10:16:02 AM Microsoft Corporation c:\winnt\system32\xolehlp.dll
msdtclog.dll 1999.8.3413.3 85.27 KB (87,312 bytes) 2/8/2000 10:16:01 AM Microsoft Corporation c:\winnt\system32\msdtclog.dll
mtxclu.dll 1999.8.3413.3 50.77 KB (51,984 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mtxclu.dll
msdtcprx.dll 1999.8.3413.7 625.77 KB (640,784 bytes) 2/8/2000 10:16:02 AM Microsoft Corporation c:\winnt\system32\msdtcprx.dll
txfaux.dll 1999.8.3413.7 370.77 KB (379,664 bytes) 2/8/2000 10:16:01 AM Microsoft Corporation c:\winnt\system32\txfaux.dll
msdtctm.dll 1999.8.3413.7 1.07 MB (1,120,528 bytes) 2/8/2000 10:16:02 AM Microsoft Corporation c:\winnt\system32\msdtctm.dll
msdtc.exe 1999.8.3413.3 6.77 KB (6,928 bytes) 2/8/2000 10:16:01 AM Microsoft Corporation c:\winnt\system32\msdtc.exe
inetpp.dll 5.00.2090.1 62.77 KB (64,272 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\inetpp.dll
win32spl.dll 5.00.2092.1 81.27 KB (83,216 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\win32spl.dll
usbmon.dll 5.00.2116.1 11.27 KB (11,536 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\usbmon.dll
tcpmon.dll 5.00.2102.1 40.77 KB (41,744 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\tcpmon.dll
pjlmon.dll 5.00.2090.1 12.77 KB (13,072 bytes) 9/9/1999 6:37:34 AM Microsoft Corporation c:\winnt\system32\pjlmon.dll
cnbjmon.dll 5.00.2090.1 43.77 KB (44,816 bytes) 9/9/1999 6:37:26 AM Microsoft Corporation c:\winnt\system32\cnbjmon.dll
localspl.dll 5.00.2119.1 245.77 KB (251,664 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\localspl.dll
spoolss.dll 5.00.2110.1 60.77 KB (62,224 bytes) 2/8/2000 9:38:46 AM Microsoft Corporation c:\winnt\system32\spoolss.dll
spoolsv.exe 5.00.2107.1 43.77 KB (44,816 bytes) 2/8/2000 9:38:46 AM Microsoft Corporation c:\winnt\system32\spoolsv.exe
rasadhlp.dll 5.00.2109.1 7.27 KB (7,440 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\rasadhlp.dll
winrnr.dll 5.00.2110.1 18.77 KB (19,216 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\winrnr.dll
rnr20.dll 5.00.2120.1 35.27 KB (36,112 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\rnr20.dll
rpcss.dll 5.00.2119.1 225.27 KB (230,672 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\rpcss.dll
svchost.exe 5.00.2090.1 7.77 KB (7,952 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\svchost.exe
iissuba.dll 5.00.0984 9.77 KB (10,000 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iissuba.dll
dssbase.dll 5.00.2120.1 140.27 KB (143,632 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\dssbase.dll
oakley.dll 5.00.2115.1 420.27 KB (430,352 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\oakley.dll
mfc42u.dll 6.00.8576.0 972.05 KB (995,384 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mfc42u.dll
polagent.dll 5.00.2110.1 102.27 KB (104,720 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\polagent.dll
scecli.dll 5.00.2112.1 101.77 KB (104,208 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\scecli.dll
esent.dll 6.0.3938.7 848.77 KB (869,136 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\esent.dll
mswsock.dll 5.00.2120.1 62.77 KB (64,272 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mswsock.dll
ntdsatq.dll 5.00.2122.1 30.77 KB (31,504 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\ntdsatq.dll

ntdsa.dll 5.00.2127.1 984.77 KB (1,008,400 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\ntdsa.dll
kdcsvc.dll 5.00.2121.1 138.77 KB (142,096 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\kdcsvc.dll
sfmapi.dll 5.00.2090.1 38.77 KB (39,696 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\sfmapi.dll
rassfm.dll 5.00.2109.1 21.27 KB (21,776 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\rassfm.dll
mpr.dll 5.00.2111.1 53.27 KB (54,544 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mpr.dll
schannel.dll 5.00.2118.1 136.77 KB (140,048 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\schannel.dll
netlogon.dll 5.00.2119.1 344.77 KB (353,040 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\netlogon.dll
kerberos.dll 5.00.2121.1 190.27 KB (194,832 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\kerberos.dll
msprivs.dll 5.00.2112.1 41.50 KB (42,496 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msprivs.dll
samsrv.dll 5.00.2124.1 352.27 KB (360,720 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\samsrv.dll
cryptdll.dll 5.00.2112.1 40.27 KB (41,232 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\cryptdll.dll
lsasrv.dll 5.00.2121.1 483.77 KB (495,376 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\lsasrv.dll
lsass.exe 5.00.2121.1 32.77 KB (33,552 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\lsass.exe
msi.dll 1.10.0816.3 1.64 MB (1,715,984 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msi.dll
appmgmts.dll 5.00.2109.1 115.77 KB (118,544 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\appmgmts.dll
wshtcpip.dll 5.00.2090.1 17.27 KB (17,680 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\wshtcpip.dll
msafd.dll 5.00.2095.1 52.27 KB (53,520 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msafd.dll
wmicore.dll 5.00.2119.1 70.27 KB (71,952 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\wmicore.dll
msgsvc.dll 5.00.2110.1 33.77 KB (34,576 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msgsvc.dll
browser.dll 5.00.2098.1 48.27 KB (49,424 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\browser.dll
alrsvc.dll 5.00.2090.1 17.77 KB (18,192 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\alrsvc.dll
trkwks.dll 5.00.2110.1 87.77 KB (89,872 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\trkwks.dll
seclogon.dll 5.00.2122.1 15.27 KB (15,632 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\seclogon.dll
psbase.dll 5.00.2090.1 110.77 KB (113,424 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\psbase.dll
cryptsvc.dll 5.00.2090.1 66.77 KB (68,368 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\cryptsvc.dll
wkssvc.dll 5.00.2120.1 91.27 KB (93,456 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\wkssvc.dll
srvsvc.dll 5.00.2117.1 79.27 KB (81,168 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\srvsvc.dll
cfgmgr32.dll 5.00.2098.1 16.77 KB (17,168 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\cfgmgr32.dll
dmserver.dll 2121.1.286.1 11.77 KB (12,048 bytes) 9/9/1999 8:00:00 PM VERITAS Software Corp. c:\winnt\system32\dmserver.dll
winsta.dll 5.00.2100.1 36.27 KB (37,136 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\winsta.dll
lmhsvc.dll 5.00.2102.1 9.27 KB (9,488 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\lmhsvc.dll
dnsrslvr.dll 5.00.2118.1 91.27 KB (93,456 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\dnsrslvr.dll
tapi32.dll 5.00.2090.1 122.27 KB (125,200 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\tapi32.dll
rasman.dll 5.00.2114.1 59.77 KB (61,200 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\rasman.dll
rasapi32.dll 5.00.2116.1 187.27 KB (191,760 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\rasapi32.dll
rtutils.dll 5.00.2109.1 43.27 KB (44,304 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\rtutils.dll
adsldpc.dll 5.00.2120.1 125.77 KB (128,784 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\adsldpc.dll
activeds.dll 5.00.2118.1 171.77 KB (175,888 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\activeds.dll
mprapi.dll 5.00.2112.1 90.77 KB (92,944 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\mprapi.dll
iphlpapi.dll 5.00.2095.2 67.27 KB (68,880 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\iphlpapi.dll
icmp.dll 5.00.2090.1 7.27 KB (7,440 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\icmp.dll
dhcpcsvc.dll 5.00.2107.1 88.77 KB (90,896 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\dhcpcsvc.dll
eventlog.dll 5.00.2090.1 43.77 KB (44,816 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\eventlog.dll
ntdsapi.dll 5.00.2120.1 55.27 KB (56,592 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\ntdsapi.dll
scesrv.dll 5.00.2112.1 220.27 KB (225,552 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\scesrv.dll
umpnpmgr.dll 5.00.2109.1 116.77 KB (119,568 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\umpnpmgr.dll
services.exe 5.00.2106.1 87.27 KB (89,360 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\services.exe
msv1_0.dll 5.00.2113.1 93.77 KB (96,016 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\msv1_0.dll
clbcatq.dll 1999.8.3413.3 494.27 KB (506,128 bytes) 2/8/2000 10:15:52 AM Microsoft Corporation c:\winnt\system32\clbcatq.dll
oleaut32.dll 2.40.4505 596.27 KB (610,576 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\oleaut32.dll
cscui.dll 5.00.2116.1 225.77 KB (231,184 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\cscui.dll
atl.dll 3.00.8449 57.56 KB (58,938 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\atl.dll
certcli.dll 5.00.2120.1 131.27 KB (134,416 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\certcli.dll
winspool.drv 5.00.2110.1 109.77 KB (112,400 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\winspool.drv
winscard.dll 5.00.2108.1 77.27 KB (79,120 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\winscard.dll
winmm.dll 5.00.2114.1 184.27 KB (188,688 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation c:\winnt\system32\winmm.dll

wlnotify.dll 5.00.2090.1 52.77 KB (54,032 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\wlnotify.dll
cscdll.dll 5.00.2122.1 97.77 KB (100,112 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\cscdll.dll
lz32.dll 5.00.2090.1 9.77 KB (10,000 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\lz32.dll
version.dll 5.00.2090.1 15.77 KB (16,144 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\version.dll
rsabase.dll 5.00.2120.1 127.77 KB (130,832 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\rsabase.dll
setupapi.dll 5.00.2126.1 551.27 KB (564,496 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\setupapi.dll
mscat32.dll 5.131.2090.1 7.77 KB (7,952 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\mscat32.dll
ole32.dll 5.00.2120.1 961.27 KB (984,336 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\ole32.dll
imagehlp.dll 5.00.2128.1 40.77 KB (41,744 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\imagehlp.dll
msasn1.dll 5.00.2090.1 50.27 KB (51,472 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\msasn1.dll
crypt32.dll 5.131.2118.1 454.77 KB (465,680 bytes) 9/9/1999 8:00:00 PM Microsoft
Corporation c:\winnt\system32\crypt32.dll
wintrust.dll 5.131.2090.1 161.27 KB (165,136 bytes) 9/9/1999 8:00:00 PM Microsoft
Corporation c:\winnt\system32\wintrust.dll
comctl32.dll 5.81 539.77 KB (552,720 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\comctl32.dll
shlwapi.dll 5.00.2919.3800 281.77 KB (288,528 bytes) 9/9/1999 8:00:00 PM Microsoft
Corporation c:\winnt\system32\shlwapi.dll
shell32.dll 5.00.2919.3800 2.24 MB (2,344,208 bytes) 9/9/1999 8:00:00 PM Microsoft
Corporation c:\winnt\system32\shell32.dll
msgina.dll 5.00.2115.1 308.27 KB (315,664 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\msgina.dll
wsock32.dll 5.00.2120.1 21.27 KB (21,776 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\wsock32.dll
dnsapi.dll 5.00.2120.1 133.77 KB (136,976 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\dnsapi.dll
wldap32.dll 5.00.2117.1 153.77 KB (157,456 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\wldap32.dll
ws2help.dll 5.00.2095.1 17.77 KB (18,192 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\ws2help.dll
ws2_32.dll 5.00.2104.1 67.77 KB (69,392 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2124.1 46.27 KB (47,376 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\samlib.dll
netrap.dll 5.00.2090.1 11.27 KB (11,536 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\netrap.dll
netapi32.dll 5.00.2120.1 295.77 KB (302,864 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\netapi32.dll
profmap.dll 5.00.2112.1 27.27 KB (27,920 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\profmap.dll
secur32.dll 5.00.2119.1 44.77 KB (45,840 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\secur32.dll
sfc.dll 5.00.2124.1 83.27 KB (85,264 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\sfc.dll
nddeapi.dll 5.00.2090.1 15.27 KB (15,632 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\nddeapi.dll
userenv.dll 5.00.2127.1 343.27 KB (351,504 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\userenv.dll
user32.dll 5.00.2120.1 392.27 KB (401,680 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\user32.dll
gdi32.dll 5.00.2115.1 228.27 KB (233,744 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\gdi32.dll
rpcrt4.dll 5.00.2128.1 440.27 KB (450,832 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\rpcrt4.dll
advapi32.dll 5.00.2120.1 337.77 KB (345,872 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\advapi32.dll
kernel32.dll 5.00.2122.1 711.77 KB (728,848 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\kernel32.dll
msvcrt.dll 6.10.8581.0 284.05 KB (290,869 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\msvcrt.dll
winlogon.exe 5.00.2116.1 171.27 KB (175,376 bytes) 9/9/1999 8:00:00 PM Microsoft
Corporation c:\winnt\system32\winlogon.exe
sfcfiles.dll 5.00.2128.1 366.77 KB (375,568 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\sfcfiles.dll
ntdll.dll 5.00.2121.1 469.27 KB (480,528 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\ntdll.dll
smss.exe 5.00.2090.1 42.77 KB (43,792 bytes) 9/9/1999 8:00:00 PM Microsoft Corporation
c:\winnt\system32\smss.exe

[Services]

Display Name Name State Start Mode Service Type Path Error Control
Start Name Tag ID
Alerter Alerter Running Auto Start Share Process c:\winnt\system32\services.exe Normal
LocalSystem0
Application Management AppMgmt Running Demand Start Share Process
c:\winnt\system32\services.exe Normal LocalSystem0
Computer Browser Browser Running Auto Start Share Process c:\winnt\system32\services.exe
Normal LocalSystem0
Indexing Service cisvc Stopped Demand Start Share Process
c:\winnt\system32\cisvc.exe Normal LocalSystem0
ClipBook ClipSrv Stopped Demand Start Own Process c:\winnt\system32\clipsrv.exe
Normal LocalSystem0
Distributed File System Dfs Running Auto Start Own Process c:\winnt\system32\dfssvc.exe
Normal LocalSystem0
DHCP Client Dhcp Running Auto Start Share Process c:\winnt\system32\services.exe
Normal LocalSystem0
Logical Disk Manager Administrative Service dmadmin Stopped Demand Start Share Process
c:\winnt\system32\dmadmin.exe/com Normal LocalSystem0
Logical Disk Manager dmserver Running Auto Start Share Process c:\winnt\system32\services.exe
Normal LocalSystem0
DNS Client Dnscache Running Auto Start Share Process c:\winnt\system32\services.exe Normal
LocalSystem0
Event Log Eventlog Running Auto Start Share Process c:\winnt\system32\services.exe Normal
LocalSystem0
COM+ Event System EventSystem Running Demand Start Share Process
c:\winnt\system32\svchost.exe-k netsvcs Normal LocalSystem0
Fax Service Fax Stopped Demand Start Own Process c:\winnt\system32\faxsvc.exe
Normal LocalSystem0
Internet Authentication Service IAS Running Auto Start Share Process
c:\winnt\system32\svchost.exe-k netsvcs Normal LocalSystem0

IIS Admin Service IISADMIN Running Auto Start Share Process
c:\winnt\system32\inetsrv\inetinfo.exe Normal LocalSystem 0
IMDB Server ImdbServer Stopped Disabled Own Process c:\winnt\system32\imdbsrv.exe
Normal LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own Process c:\winnt\system32\ismserv.exe
Normal LocalSystem 0
Kerberos Key Distribution Center kdc Stopped Disabled Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Server lanmanserver Running Auto Start Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
Workstation lanmanworkstation Running Auto Start Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
License Logging Service LicenseService Running Auto Start Own Process
c:\winnt\system32\llssrv.exe Normal LocalSystem 0
TCP/IP NetBIOS Helper Service LmHosts Running Auto Start Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Messenger Messenger Running Auto Start Share Process c:\winnt\system32\services.exe Normal
LocalSystem 0
NetMeeting Remote Desktop Sharing mnmsrvc Stopped Demand Start Own Process
c:\winnt\system32\mnmsrvc.exe Normal LocalSystem 0
Distributed Transaction Coordinator MSDTC Running Auto Start Own Process
c:\winnt\system32\msdtc.exe Normal LocalSystem 0
Windows Installer MSIServer Stopped Demand Start Share Process
c:\winnt\system32\msiexec.exe/v Normal LocalSystem 0
Network DDE NetDDE Stopped Demand Start Share Process
c:\winnt\system32\netdde.exe Normal LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped Demand Start Share Process
c:\winnt\system32\netdde.exe Normal LocalSystem 0
Net Logon Netlogon Stopped Demand Start Share Process c:\winnt\system32\lsass.exe
Normal LocalSystem 0
Network Connections Netman Running Demand Start Share Process
c:\winnt\system32\svchost.exe-k netsvcs Normal LocalSystem 0
File Replication NtFrs Stopped Demand Start Own Process
c:\winnt\system32\ntfrs.exe Ignore LocalSystem 0
NT LM Security Support Provider NtLmSsp Stopped Demand Start Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Removable Storage NtmsSvc Running Auto Start Share Process c:\winnt\system32\svchost.exe-k
netsvcs Normal LocalSystem 0
Plug and Play PlugPlay Running Auto Start Share Process c:\winnt\system32\services.exe
Normal LocalSystem 0
IPSEC Policy Agent PolicyAgent Running Auto Start Share Process c:\winnt\system32\lsass.exe
Normal LocalSystem 0
Protected Storage ProtectedStorage Running Auto Start Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto Stopped Demand Start Share Process
c:\winnt\system32\svchost.exe-k netsvcs Normal LocalSystem 0
Remote Access Connection Manager RasMan Stopped Demand Start Share Process
c:\winnt\system32\svchost.exe-k netsvcs Normal LocalSystem 0
Routing and Remote Access RemoteAccess Stopped Disabled Share Process
c:\winnt\system32\svchost.exe-k netsvcs Normal LocalSystem 0
Remote Registry Service RemoteRegistry Running Auto Start Own Process
c:\winnt\system32\regsvc.exe Normal LocalSystem 0
Remote Procedure Call (RPC) Locator RpcLocator Stopped Demand Start Own Process
c:\winnt\system32\locator.exe Normal LocalSystem 0
Remote Procedure Call (RPC) RpcSs Running Auto Start Share Process
c:\winnt\system32\svchost-k rpcss Normal LocalSystem 0
QoS Admission Control (RSVP) RSVP Running Auto Start Own Process
c:\winnt\system32\rsvp.exe-s Normal LocalSystem 0
Security Accounts Manager SamSs Running Auto Start Share Process
c:\winnt\system32\lsass.exe Normal LocalSystem 0
Smart Card Helper SCardDrv Stopped Demand Start Share Process
c:\winnt\system32\scardsvr.exe Ignore LocalSystem 0
Smart Card SCardSvr Stopped Demand Start Share Process c:\winnt\system32\scardsvr.exe
Ignore LocalSystem 0
Task Scheduler Schedule Running Auto Start Share Process c:\winnt\system32\mstask.exe
Normal LocalSystem 0
RunAs Service seclogon Running Auto Start Share Process c:\winnt\system32\services.exe
Ignore LocalSystem 0
System Event Notification SENS Running Auto Start Share Process
c:\winnt\system32\svchost.exe-k netsvcs Normal LocalSystem 0
Internet Connection Sharing SharedAccess Stopped Demand Start Share Process
c:\winnt\system32\svchost.exe-k netsvcs Normal LocalSystem 0
Simple TCP/IP Services SimpTcp Running Auto Start Share Process c:\winnt\system32\tcpsvcs.exe
Normal LocalSystem 0
Print Spooler Spooler Running Auto Start Own Process c:\winnt\system32\spoolsv.exe
Normal LocalSystem 0
Performance Logs and Alerts SysmonLog Stopped Demand Start Own Process
c:\winnt\system32\smlogsvc.exe Normal LocalSystem 0
Telephony TapiSrv Stopped Demand Start Share Process c:\winnt\system32\svchost.exe-k
tapisrv Normal LocalSystem 0
Terminal Services TermService Stopped Disabled Own Process c:\winnt\system32\termsrv.exe
Normal LocalSystem 0
Telnet TlntSvr Stopped Demand Start Own Process c:\winnt\system32\tlntsvr.exe
Normal LocalSystem 0
Distributed Link Tracking Server TrkSvr Stopped Demand Start Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Distributed Link Tracking Client TrkWks Running Auto Start Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
Uninterruptible Power Supply UPS Stopped Demand Start Own Process
c:\winnt\system32\ups.exe Normal LocalSystem 0
Utility Manager UtilMan Stopped Demand Start Own Process
c:\winnt\system32\utilman.exe Normal LocalSystem 0
Windows Time W32Time Stopped Demand Start Share Process
c:\winnt\system32\services.exe Normal LocalSystem 0
World Wide Web Publishing Service W3SVC Running Auto Start Share Process
c:\winnt\system32\inetsrv\inetinfo.exe Normal LocalSystem 0
Windows Management Instrumentation WinMgmt Running Demand Start Own Process
c:\winnt\system32\wbem\winmgmt.exe Ignore LocalSystem 0
Windows Management Instrumentation Driver Extensions Wmi Running Demand Start Share
Process c:\winnt\system32\services.exe Normal LocalSystem 0

[Program Groups]

Group Name Name User Name
Accessories Default User:Accessories Default User
Accessories\Accessibility Default User:Accessories\Accessibility Default User
Accessories\Entertainment Default User:Accessories\Entertainment Default User
Accessories\System Tools Default User:Accessories\System Tools Default User
Startup Default User:Startup Default User
Accessories All Users:Accessories All Users
Accessories\Accessibility All Users:Accessories\Accessibility All Users
Accessories\Communications All Users:Accessories\Communications All Users

Accessories\Entertainment All Users:Accessories\Entertainment All Users
Accessories\Games All Users:Accessories\Games All Users
Accessories\Microsoft Script Debugger All Users:Accessories\Microsoft Script Debugger All Users
Accessories\System Tools All Users:Accessories\System Tools All Users
Administrative Tools All Users:Administrative Tools All Users
Microsoft SQL Server 7.0 All Users:Microsoft SQL Server 7.0 All Users
Startup All Users:Startup All Users
Accessories SQLCLIENT40\Administrator:Accessories SQLCLIENT40\Administrator
Accessories\Accessibility SQLCLIENT40\Administrator:Accessories\Accessibility SQLCLIENT40\Administrator
Accessories\Entertainment SQLCLIENT40\Administrator:Accessories\Entertainment
SQLCLIENT40\Administrator
Accessories\System Tools SQLCLIENT40\Administrator:Accessories\SystemTools
SQLCLIENT40\Administrator
Administrative Tools SQLCLIENT40\Administrator:AdministrativeTools SQLCLIENT40\Administrator
Startup SQLCLIENT40\Administrator:StartupSQLCLIENT40\Administrator

[Startup Programs]

Program Command User Name Location
No startup program information

[OLE Registration]

Object Local Server
Sound (OLE2) sndrec32.exe
Media Clip mplay32.exe
Video Clip mplay32.exe /avi
MIDI Sequence mplay32.exe /mid
Sound Not Available
Media Clip Not Available
Image Document "C:\Program Files\Windows NT\Accessories\ImageVue\KodakImg.exe"
WordPad Document "%ProgramFiles%\Windows NT\Accessories\WORDPAD.EXE"
Windows Media Services DRM Storage object Not Available
Bitmap Image mspaint.exe

[Internet Explorer 5]

[Following are sub-categories of this main category]

[Summary]

Item Value
Version 5.00.2919.3800
Build 52919.3800
Product ID 50293-270-0207633-13548
Application Path C:\Program Files\Internet Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 56-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File Version Size Date Path Company

advapi32.dll 5.0.2120.1 338 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
advpack.dll 5.0.2919.3800 87 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
browselc.dll 5.0.2919.3800 35 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
browseui.dll 5.0.2919.3800 792 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
ckcnv.exe 5.0.2120.1 9 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
comctl32.dll 5.81.2919.3800 540 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
crypt32.dll 5.131.2118.1 455 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
enhsig.dll <File Missing> Not Available Not Available Not Available Not Available
iemigrat.dll <File Missing> Not Available Not Available Not Available Not Available
iesetup.dll 5.0.2919.3800 57 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
iexplore.exe 5.0.2919.3800 59 KB 9/9/1999 7:00:00 PM C:\Program Files\Internet Explorer Microsoft Corporation
imagehlp.dll 5.0.2128.1 41 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
imghelp.dll <File Missing> Not Available Not Available Not Available Not Available
inseng.dll 5.0.2919.3800 71 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
jobexec.dll 5.0.0.1 47 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
jscript.dll 5.1.0.4411 476 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
jsproxy.dll 5.0.2919.3800 13 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
msaahtml.dll <File Missing> Not Available Not Available Not Available Not Available
mshtml.dll 5.0.2919.3800 2301 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
msjava.dll 5.0.3229.0 918 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
msoss.dll <File Missing> Not Available Not Available Not Available Not Available
msxml.dll 5.0.2919.3800 509 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
occache.dll 5.0.2919.3800 86 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
ole32.dll 5.0.2120.1 961 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
oleaut32.dll 2.40.4505.1 596 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
olepro32.dll 5.0.4505.1 156 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
rsabase.dll 5.0.2120.1 128 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
rsaenh.dll <File Missing> Not Available Not Available Not Available Not Available
rsapi32.dll <File Missing> Not Available Not Available Not Available Not Available
rsasig.dll <File Missing> Not Available Not Available Not Available Not Available
schannel.dll 5.0.2118.0 137 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
shdoc401.dll <File Missing> Not Available Not Available Not Available Not Available
shdocvw.dll 5.0.2919.3800 1078 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation

shell32.dll 5.0.2919.3800 2289 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
shlwapi.dll 5.0.2919.3800 282 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
url.dll 5.0.2919.3800 82 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
urlmon.dll 5.0.2919.3800 427 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
vbscript.dll 5.1.0.4411 428 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
webcheck.dll 5.0.2919.3800 252 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
win.com 5.0.2090.1 24 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
wininet.dll 5.0.2919.3800 457 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
winsock.dll 3.10.0.103 3 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
wintrust.dll 5.131.2090.1 161 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
wsock.vxd <File Missing> Not Available Not Available Not Available Not Available
wsock32.dll 5.0.2120.1 21 KB 9/9/1999 7:00:00 PM C:\WINNT\system32 Microsoft Corporation
wsock32n.dll <File Missing> Not Available Not Available Not Available Not Available

[Connectivity]

Item Value
Connection Preference Never dial
EnableHttp1.1 1
ProxyHttp1.1 0

LAN Settings

AutoConfigProxy wininet.dll
AutoProxyDetectMode Enabled
AutoConfigURL
Proxy Disabled
ProxyServer
ProxyOverride

[Cache]

[Following are sub-categories of this main category]

[Summary]

Item Value
Page Refresh Type Automatic
Temporary Internet Files Folder C:\Documents and Settings\Administrator.SQLCLIENT40\LocalSettings\Temporary Internet Files
Total Disk Space 4290 MB
Available Disk Space 2551 MB
Maximum Cache Size 134 MB
Available Cache Size 134 MB

[List of Objects]

Program FileStatus CodeBase
No cached object information available

[Content]

[Following are sub-categories of this main category]

[Summary]

Item Value
Content Advisor Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm
Administrator Administrator 2/8/2000 to 1/14/2100 sha1RSA

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone Security Level
Local intranet Medium-low
Trusted sites Low
Internet Medium
Restricted sites High

COM+ Settings

TPCC.AllTxns:
Activation:
Enable Object Pooling selected
Minimum Pool Size: 64
Maximum Pool Size: 64
Creating Timeout: 60,000
Enable Object Construction
Enable Just in Time Activation
Concurrency:
Concurrency Required

TPCC Application Registry Parameters

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="c:\\inetpub\\wwwroot\\"
"NumberOfDeliveryThreads"=dword:00000007

```
"MaxConnections"=dword:00001f40
"MaxPendingDeliveries"=dword:000003e8
"DB_Protocol"="DBLIB"
"TxnMonitor"="COM"
"DbServer"="ibmserv3"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"
```

Microsoft Internet Information Service Registry Parameters

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,00,00
"PoolThreadLimit"=dword:000000be
"ThreadTimeout"=dword:00015180

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000842
"Last Help"=dword:00000843
"First Counter"=dword:00000802
```

World Wide Web Service Registry Parameters

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,4e,00,54,00,5c,00,53,00,\
  79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,65,00,74,00,73,\
  00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e,00,66,00,6f,00,2e,00,\
  65,00,78,00,65,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,00,49,00,53,00,41,00,44,00,4d,00,49,00,4e,00,00,00,\
  00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and administration through the Internet Information Services snap-in."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP]
"NOTE"="This is for backward compatibility only."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINNT\\System32\\inetsrv"
"CertMapList"="C:\\WINNT\\System32\\inetsrv\\iiscrmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\\WINNT\\System32\\LogFiles"
"AcceptExOutstanding"=dword:00000028

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactor
y]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFact
ory]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots]
"/"="c:\\inetpub\\wwwroot,,205"
"/Scripts"="c:\\inetpub\\scripts,,1"
"/IISAdmin"="C:\\WINNT\\System32\\inetsrv\\iisadmin,,1"
"/IISSamples"="c:\\inetpub\\iissamples,,1"
"/MSADC"="c:\\program files\\common files\\system\\msadc,,1"
"/IISHelp"="c:\\winnt\\help\\iishelp,,1"
"/_vti_bin"="C:\\Program Files\\Common Files\\Microsoft Shared\\Web Server Extensions\\40\\isapi,,1"
"/Rpc"="C:\\WINNT\\System32\\RpcProxy,,1"
"/Printers"="C:\\WINNT\\web\\printers,,201"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Performance]
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008f2
"Last Help"=dword:000008f3
"First Counter"=dword:00000850
"First Help"=dword:00000851

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

RTE Input Parameters

Profile: 3240wh-36segments
File Path: E:\benchcrf\3240wh-36segments.pro
Version: 1.0.1

Number of Engines: 36

Name: DRIVER1
Description: rte11_toclient
Directory: c:\benchcrf\logs\driver1.log
Machine: rte10
Parameter Set: PARAM2
Index: 0
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER11594390
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER10
Description: rte24_toclient
Directory: c:\benchcrf\logs\driver10.log
Machine: rte20
Parameter Set: PARAM2
Index: 900000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER101942843
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER11
Description: rte25_toclient
Directory: c:\benchcrf\logs\river11.log
Machine: rte20
Parameter Set: PARAM2
Index: 1000000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER111969046
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER12
Description: rte26_toclient
Directory: c:\benchcrf\logs\driver12.log
Machine: rte20
Parameter Set: PARAM2
Index: 1100000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER121997234
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER13
Description: rte31_toclient
Directory: c:\benchcrf\logs\driver13.log
Machine: rte30
Parameter Set: PARAM2
Index: 1200000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER132028921
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER14
Description: rte32_toclient
Directory: c:\benchcrf\logs\driver14.log
Machine: rte30
Parameter Set: PARAM2
Index: 1300000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER142053484
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER15
Description: rte33_toclient
Directory: c:\benchcrf\logs\driver15.log
Machine: rte30
Parameter Set: PARAM2
Index: 1400000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER152082187
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER16
Description: rte34_toclient
Directory: c:\benchcrf\logs\driver16.log
Machine: rte30
Parameter Set: PARAM2

Index: 1500000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER162114078
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER17
Description: rte35_toclient
Directory: c:\benchcrf\logs\driver17.log
Machine: rte30
Parameter Set: PARAM2
Index: 1600000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER172146265
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER18
Description: rte36_toclient
Directory: c:\benchcrf\logs\driver18.log
Machine: rte30
Parameter Set: PARAM2
Index: 1700000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER182173968
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER19
Description: rte41_toclient
Directory: c:\benchcrf\logs\driver19.log
Machine: rte40
Parameter Set: PARAM2
Index: 1800000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER192201718
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER2
Description: rte12_toclient
Directory: c:\benchcrf\logs\driver2.log
Machine: rte10
Parameter Set: PARAM2

Index: 100000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER21672640
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER20
Description: rte42_toclient
Directory: c:\benchcrf\logs\driver20.log
Machine: rte40
Parameter Set: PARAM2
Index: 1900000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER202229015
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER21
Description: rte43_toclient
Directory: c:\benchcrf\logs\driver21.log
Machine: rte40
Parameter Set: PARAM2
Index: 2000000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER212261093
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER22
Description: rte44_toclient
Directory: c:\benchcrf\logs\driver22.log
Machine: rte40
Parameter Set: PARAM2
Index: 2100000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER222287671
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER23
Description: rte45_toclient
Directory: c:\benchcrf\logs\driver23.log
Machine: rte40
Parameter Set: PARAM2

Index: 80000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER232312734
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER24
Description: rte46_toclient
Directory: c:\benchcrf\logs\driver24.log
Machine: rte40
Parameter Set: PARAM2
Index: 70000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER242405515
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER25
Description: rte51_toclient
Directory: c:\benchcrf\logs\driver25.log
Machine: rte50
Parameter Set: PARAM2
Index: 60000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER252459453
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER26
Description: rte52_toclient
Directory: c:\benchcrf\logs\driver26.log
Machine: rte50
Parameter Set: PARAM2
Index: 50000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER262492625
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER27
Description: rte53_toclient
Directory: c:\benchcrf\logs\driver27.log
Machine: rte50
Parameter Set: PARAM2

Index: 40000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER272532578
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER28
Description: rte54_toclient
Directory: c:\benchcrf\logs\driver28.log
Machine: rte50
Parameter Set: PARAM2
Index: 30000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER282567687
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER29
Description: rte55_toclient
Directory: c:\benchcrf\logs\driver29.log
Machine: rte50
Parameter Set: PARAM2
Index: 20000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER292610578
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER3
Description: rte13_toclient
Directory: c:\benchcrf\logs\driver3.log
Machine: rte10
Parameter Set: PARAM2
Index: 200000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER31704859
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER30
Description: rte56_toclient
Directory: c:\benchcrf\logs\driver30.log
Machine: rte50
Parameter Set: PARAM2

Index: 10000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER302655687
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER31
Description: rte61_toclient
Directory: c:\benchcrf\logs\driver31.log
Machine: rte60
Parameter Set: PARAM2
Index: 110000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER312701687
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER32
Description: rte62_toclient
Directory: c:\benchcrf\logs\driver32.log
Machine: rte60
Parameter Set: PARAM2
Index: 120000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER322763046
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER33
Description: rte63_toclient
Directory: c:\benchcrf\logs\driver33.log
Machine: rte60
Parameter Set: PARAM2
Index: 130000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER332804015
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER34
Description: rte64_toclient
Directory: c:\benchcrf\logs\driver34.log
Machine: rte60
Parameter Set: PARAM2
Index: 140000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER342838109
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER35
Description: rte65_toclient
Directory: c:\benchcrf\logs\driver35.log
Machine: rte60
Parameter Set: PARAM2
Index: 150000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER352874468
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER36
Description: rte66_toclient
Directory: c:\benchcrf\logs\driver36.log
Machine: rte60
Parameter Set: PARAM2
Index: 160000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER362905296
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER4
Description: rte14_toclient
Directory: c:\benchcrf\logs\driver4.log
Machine: rte10
Parameter Set: PARAM2
Index: 300000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER41739734
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER5
Description: rte15_toclient
Directory: c:\benchcrf\logs\driver5.log
Machine: rte10
Parameter Set: PARAM2

Index: 400000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER51768187
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER6
Description: rte16_toclient
Directory: c:\benchcrf\logs\driver6.log
Machine: rte10
Parameter Set: PARAM2
Index: 500000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER61803546
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER7
Description: rte21_toclient
Directory: c:\benchcrf\logs\driver7.log
Machine: rte20
Parameter Set: PARAM2
Index: 600000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER71838546
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Name: DRIVER8
Description: rte22_toclient
Directory: c:\benchcrf\logs\driver8.log
Machine: rte20
Parameter Set: PARAM2
Index: 700000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER81865281
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 1

Name: DRIVER9
Description: rte23_toclient
Directory: c:\benchcrf\logs\driver9.log
Machine: rte20
Parameter Set: PARAM2
Index: 800000000
Seed: 28630
Configured Users: 900
Pipe Name: DRIVER91899625
Connect Rate: 500
Start Rate: 0
CLIENT_NURAND: 208
CPU: 0

Number of User groups: 36

Driver Engine: DRIVER1
IIS Server: client11_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1 - 90
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: client14_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 271 - 360
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: client15_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 361 - 450
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER25
IIS Server: client51_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2161 - 2250
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900

District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: client16_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 451 - 540
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER7
IIS Server: client21_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 541 - 630
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: client22_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 631 - 720
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER19
IIS Server: client41_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1621 - 1710
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER9
IIS Server: client23_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 721 - 810
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: client26_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 991 - 1080
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: client31_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1081 - 1170
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER31
IIS Server: client61_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2701 - 2790
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: client32_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1171 - 1260
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER15

IIS Server: client33_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1261 - 1350
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER16
IIS Server: client34_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1351 - 1440
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER26
IIS Server: client52_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2251 - 2340
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER17
IIS Server: client35_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1441 - 1530
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: client12_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 91 - 180
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER20
IIS Server: client42_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1711 - 1800
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER21
IIS Server: client43_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1801 - 1890
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER22
IIS Server: client44_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1891 - 1980
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER23
IIS Server: client45_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1981 - 2070
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER32
IIS Server: client62_torte
SQL Server: ibmserv3
User: sa
Protocol: Html

w_id Range: 2791 - 2880
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER24
IIS Server: client46_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2071 - 2160
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: client24_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 811 - 900
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER27
IIS Server: client53_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2341 - 2430
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER28
IIS Server: client54_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2431 - 2520
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER29
IIS Server: client55_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2521 - 2610
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: client13_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 181 - 270
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER30
IIS Server: client56_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2611 - 2700
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER18
IIS Server: client36_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 1531 - 1620
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER33
IIS Server: client63_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2881 - 2970
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900

District id: 1
Scale Down: No

Driver Engine: DRIVER34
IIS Server: client64_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 2971 - 3060
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER35
IIS Server: client65_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 3061 - 3150
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: client25_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 901 - 990
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER36
IIS Server: client66_torte
SQL Server: ibmserv3
User: sa
Protocol: Html
w_id Range: 3151 - 3240
w_id Max Warehouse: 3240
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Number of Parameter Sets: 2

PARAM2
first try

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	44.80	12.03	18.02	0.10	5.00	0.10
Payment	43.05	12.03	3.02	0.10	5.00	0.10
Delivery	4.05	5.03	2.02	0.10	5.00	0.10
Stock Level	4.05	5.03	2.02	0.10	20.00	0.10
Order Status	4.05	10.05	2.02	0.10	5.00	0.10

~Default
Default Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	10.00	12.05	18.01	0.10	5.00	0.10
Payment	10.00	12.05	3.01	0.10	5.00	0.10
Delivery	1.00	5.05	2.01	0.10	5.00	0.10
Stock Level	1.00	5.05	2.01	0.10	20.00	0.10
Order Status	1.00	10.05	2.01	0.10	5.00	0.10

Appendix D: Hardware/Software Configuration Utility

Server Hardware

Microsoft Diagnostics Report For \\IBMSERV3

OS Version Report

Microsoft (R) Windows NT (TM) Server
Version 4.0 (Build 1381: Service Pack 6) x86 Multiprocessor Free
Registered Owner: TPC-C, IBM
Product Number: 70234-012-0123456-18037

System Report

System: AT/AT COMPATIBLE
Hardware Abstraction Layer: MPS 1.4 - APIC platform
BIOS Date: 11/30/99
BIOS Version: IBM BIOS Ver 1.0

Processor list:
0: x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
1: x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
2: x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
3: x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
4: x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
5: x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
6: x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz
7: x86 Family 6 Model 7 Stepping 3 GenuineIntel ~550 Mhz

Video Display Report

BIOS Date: 09/02/99
BIOS Version: S3 86C366 Video BIOS. Version 2.0B.05-C5.03.04a 83MHz
S3 86C366 Video BIOS. Version 2.0B.05-C5.03.04a 83MHz
S3 86C366 Video BIOS. Version 2.0B.05-C5.03.04a 83MHz

Adapter:
Setting: 800 x 600 x 16
Hardware Default Refresh
Type: vga compatible display adapter
String: <unavailable>
Memory:
Chip Type: <unavailable>
DAC Type: <unavailable>
Driver:
Vendor: Microsoft Corporation
File(s): vga.sys, vga.dll
Version: 4.00, 4.0.0

Drives Report

C:\ (Local - NTFS) Total: 4,192,933 KB, Free: 2,809,820 KB
Z:\ (Local - NTFS) backup2 Total: 325,816,316 KB, Free: 73,838,324 KB

Memory Report

Handles: 2,909
Threads: 230
Processes: 23

Physical Memory (K)
Total: 3,931,528
Available: 1,865,560
File Cache: 15,620

Services Report

Alerter Running (Automatic)
Computer Browser Running (Automatic)
Disk Array Monitor Running (Automatic)
EventLog (Event log) Running (Automatic)
Gopher Publishing Service Running (Automatic)
Server Running (Automatic)
Workstation (NetworkProvider) Running (Automatic)
License Logging Service Running (Automatic)
TCP/IP NetBIOS Helper Running (Automatic)
Messenger Running (Automatic)
MSDTC (MS Transactions) Running (Automatic)
FTP Publishing Service Running (Automatic)
NT LM Security Support Provider Running (Manual)
Plug and Play (PlugPlay) Running (Automatic)
Protected Storage Running (Automatic)
Remote Procedure Call (RPC) Service Running (Automatic)
Spooler (SpoolerGroup) Running (Automatic)
World Wide Web Publishing Service Running (Automatic)

Drivers Report

AFD Networking Support Environment (TDI) Running (Automatic)
aic78u2 (SCSI miniport) Running (Boot)
atapi (SCSI miniport) Running (Boot)
Beep (Base) Running (System)
Cdfs (File system) Running (Disabled)
Cdrom (SCSI CDROM Class) Running (System)
Disk (SCSI Class) Running (Boot)
Floppy (Primary disk) Running (System)
i8042 Keyboard and PS/2 Mouse Port Driver (Keyboard Port) Running (System)

IBM ??? NDIS Driver (NDIS) Running (Automatic)
intlfxsr (Base) Running (Boot)
Keyboard Class Driver (Keyboard Class) Running (System)
KSecDD (Base) Running (System)
Mouse Class Driver (Pointer Class) Running (System)
mraid (Primary disk) Running (Boot)
Msfs (File system) Running (System)
Mup (Network) Running (Manual)
Microsoft NDIS System Driver (NDIS) Running (System)
NetBIOS Interface (NetBIOSGroup) Running (Manual)
WINS Client(TCP/IP) (PNP_TDI) Running (Automatic)
Npfs (File system) Running (System)
Ntfs (File system) Running (Disabled)
Null (Base) Running (System)
Parallel (Extended base) Running (Automatic)
Parport (Parallel arbitrator) Running (Automatic)
ParVdm (Extended base) Running (Automatic)
ql2100 (SCSI Miniport) Running (Boot)
Rdr (Network) Running (Manual)
Scsiscan (SCSI Class) Running (System)
Serial (Extended base) Running (Automatic)
Srv (Network) Running (Manual)
symarray (SCSI Class) Running (Boot)
TCP/IP Service (PNP_TDI) Running (Automatic)
VgaSave (Video Save) Running (System)

IRQ and Port Report

Devices	Vector	Level	Affinity
MPS 1.4 - APIC platform	8	8	0x000000ff
MPS 1.4 - APIC platform	0	0	0x000000ff
MPS 1.4 - APIC platform	1	1	0x000000ff
MPS 1.4 - APIC platform	2	2	0x000000ff
MPS 1.4 - APIC platform	3	3	0x000000ff
MPS 1.4 - APIC platform	4	4	0x000000ff
MPS 1.4 - APIC platform	5	5	0x000000ff
MPS 1.4 - APIC platform	6	6	0x000000ff
MPS 1.4 - APIC platform	7	7	0x000000ff
MPS 1.4 - APIC platform	8	8	0x000000ff
MPS 1.4 - APIC platform	9	9	0x000000ff
MPS 1.4 - APIC platform	10	10	0x000000ff
MPS 1.4 - APIC platform	11	11	0x000000ff
MPS 1.4 - APIC platform	12	12	0x000000ff
MPS 1.4 - APIC platform	13	13	0x000000ff
MPS 1.4 - APIC platform	14	14	0x000000ff
MPS 1.4 - APIC platform	15	15	0x000000ff
MPS 1.4 - APIC platform	16	16	0x000000ff
MPS 1.4 - APIC platform	17	17	0x000000ff
MPS 1.4 - APIC platform	18	18	0x000000ff
MPS 1.4 - APIC platform	19	19	0x000000ff
MPS 1.4 - APIC platform	20	20	0x000000ff
MPS 1.4 - APIC platform	21	21	0x000000ff
MPS 1.4 - APIC platform	22	22	0x000000ff
MPS 1.4 - APIC platform	23	23	0x000000ff
MPS 1.4 - APIC platform	24	24	0x000000ff
MPS 1.4 - APIC platform	25	25	0x000000ff
MPS 1.4 - APIC platform	26	26	0x000000ff
MPS 1.4 - APIC platform	27	27	0x000000ff
MPS 1.4 - APIC platform	28	28	0x000000ff
MPS 1.4 - APIC platform	29	29	0x000000ff
MPS 1.4 - APIC platform	30	30	0x000000ff
MPS 1.4 - APIC platform	31	31	0x000000ff
MPS 1.4 - APIC platform	32	32	0x000000ff
MPS 1.4 - APIC platform	33	33	0x000000ff
MPS 1.4 - APIC platform	34	34	0x000000ff
MPS 1.4 - APIC platform	35	35	0x000000ff
MPS 1.4 - APIC platform	36	36	0x000000ff
MPS 1.4 - APIC platform	37	37	0x000000ff
MPS 1.4 - APIC platform	38	38	0x000000ff
MPS 1.4 - APIC platform	39	39	0x000000ff
MPS 1.4 - APIC platform	40	40	0x000000ff
MPS 1.4 - APIC platform	41	41	0x000000ff
MPS 1.4 - APIC platform	42	42	0x000000ff
MPS 1.4 - APIC platform	43	43	0x000000ff
MPS 1.4 - APIC platform	44	44	0x000000ff
MPS 1.4 - APIC platform	45	45	0x000000ff
MPS 1.4 - APIC platform	46	46	0x000000ff
MPS 1.4 - APIC platform	47	47	0x000000ff
MPS 1.4 - APIC platform	61	61	0x000000ff
MPS 1.4 - APIC platform	65	65	0x000000ff
MPS 1.4 - APIC platform	80	80	0x000000ff
MPS 1.4 - APIC platform	193	193	0x000000ff
MPS 1.4 - APIC platform	225	225	0x000000ff
MPS 1.4 - APIC platform	253	253	0x000000ff
MPS 1.4 - APIC platform	254	254	0x000000ff
MPS 1.4 - APIC platform	255	255	0x000000ff
i8042prt	1	1	0xffffffff
i8042prt	12	12	0xffffffff
Serial	4	4	0x00000000
Serial	3	3	0x00000000
Floppy	6	6	0x00000000
IBMFEPCI	12	12	0x00000000
aic78u2	56	56	0x00000000
aic78u2	56	56	0x00000000
atapi	0	14	0x00000000
ql2100	40	40	0x00000000

Devices	Physical Address	Length
MPS 1.4 - APIC platform	0x00000000	0x0000000010
MPS 1.4 - APIC platform	0x00000020	0x0000000002
MPS 1.4 - APIC platform	0x00000040	0x0000000004
MPS 1.4 - APIC platform	0x00000048	0x0000000004
MPS 1.4 - APIC platform	0x00000061	0x0000000001
MPS 1.4 - APIC platform	0x00000070	0x0000000002
MPS 1.4 - APIC platform	0x00000080	0x0000000010
MPS 1.4 - APIC platform	0x00000092	0x0000000001
MPS 1.4 - APIC platform	0x000000a0	0x0000000002
MPS 1.4 - APIC platform	0x000000c0	0x0000000010

```
MPS 1.4 - APIC platform        0x000000f0  0x0000000010
i8042prt                       0x00000060  0x0000000001
i8042prt                       0x00000064  0x0000000001
Parport                        0x00000378  0x0000000003
Serial                         0x000003f8  0x0000000007
Serial                         0x000002f8  0x0000000007
Floppy                         0x000003f0  0x0000000006
Floppy                         0x000003f7  0x0000000001
IBMFEPCI                           0x0000a000  0x000000001e
aic78u2                        0x00002300  0x0000000100
aic78u2                        0x00002400  0x0000000100
atapi                          0x000001f0  0x0000000008
atapi                          0x000003f6  0x0000000001
ql2100                         0x00002200  0x0000000100
VgaSave                            0x000003b0  0x000000000c
VgaSave                            0x000003c0  0x0000000020
VgaSave                            0x000001ce  0x0000000002


DMA and Memory Report
-------------------------------------------------------------------------------
Devices                        Channel    Port
-------------------------------------------------------------------------------
Floppy                          2     0
-------------------------------------------------------------------------------
Devices                        Physical Address  Length
-------------------------------------------------------------------------------
MPS 1.4 - APIC platform        0xfec00000  0x00000400
MPS 1.4 - APIC platform        0xfee00000  0x00000400
IBMFEPCI                           0xfb3ff000  0x0000001e
aic78u2                        0xf9bfd000  0x00001000
aic78u2                        0xf9bfc000  0x00001000
ql2100                         0xf9bfe000  0x00001000
VgaSave                            0x000a0000  0x00020000


Environment Report
-------------------------------------------------------------------------------


System Environment Variables
   ComSpec=C:\WINNT\system32\cmd.exe
   Os2LibPath=C:\WINNT\system32\os2\dll;
   Path=C:\WINNT\system32;C:\WINNT;C:\MSSQL7\BINN
   windir=C:\WINNT
   OS=Windows_NT
   PROCESSOR_ARCHITECTURE=x86
   PROCESSOR_LEVEL=6
   PROCESSOR_IDENTIFIER=x86 Family 6 Model 7 Stepping 3, GenuineIntel
   PROCESSOR_REVISION=0703
   NUMBER_OF_PROCESSORS=8
   ND_HOME=C:\Program Files\SYMsm
   ND_PATH=C:\Program Files\SYMsm

Environment Variables for Current User
   TEMP=C:\TEMP
   TMP=C:\TEMP



Network Report
-------------------------------------------------------------------------------
Your Access Level: Admin & Local
Workgroup or Domain: WORKGROUP
Network Version: 4.0
LanRoot: WORKGROUP
Logged On Users: 1
Current User (1): Administrator
  Logon Domain: IBMSERV3
  Logon Server: IBMSERV3

Transport: NetBT_IBMFEPCI1, 00-04-AC-36-44-FE, VC's: 0, Wan: Wan

Character Wait: 3,600
Collection Time: 250
Maximum Collection Count: 16
Keep Connection: 600
Maximum Commands: 5
Session Time Out: 45
Character Buffer Size: 512
Maximum Threads: 17
Lock Quota: 6,144
Lock Increment: 10
Maximum Locks: 500
Pipe Increment: 10
Maximum Pipes: 500
Cache Time Out: 40
Dormant File Limit: 45
Read Ahead Throughput: 4,294,967,295
Mailslot Buffers: 3
Server Announce Buffers: 20
Illegal Datagrams: 5
Datagram Reset Frequency: 60
Bytes Received: 269
SMB's Received: 3
```

Server Software

```
System Name:
Dictionary Name: C:\WNETFIN\default.sid
===============================================================
Product Name: Notepad
Vendor Name: Microsoft
Location: C:\WINNT
Application Type: Word Processing
Description: Windows Notepad Editor
===============================================================
Product Name: Netfinity Services for Windows (32 bit) v. 5.20.x
```

Vendor Name: IBM
Version: 5.20
Revision: x
Location: C:\WNETFIN
Application Type: System Management
Description: Network Management Suite
==========
Product Name: Netfinity Manager for Windows (32 bit) v. 5.20.x
Vendor Name: IBM
Version: 5.20
Revision: x
Location: C:\WNETFIN
Application Type: System Management
Description: Network Management Suite
==========
Product Name: Phone Dialer
Vendor Name: Microsoft
Location: C:\Program Files\Windows NT
Application Type: Communications
Description: Windows Telephone Dialer
==========
Product Name: Phone Dialer
Vendor Name: Microsoft
Location: C:\WINNT\$NtServicePackUninstall$
Application Type: Communications
Description: Windows Telephone Dialer
==========
Product Name: Backup
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: System Management
Description: Archive Files to Disk or Tape
==========
Product Name: CD Player
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Multimedia
Description: Windows CD Player
==========
Product Name: Character Map
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Word Processing
Description: Windows Character Map
==========
Product Name: Clipboard Viewer
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Word Processing
Description: Windows Clipboard Viewer
==========
Product Name: Notepad
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Word Processing
Description: Windows Notepad Editor
==========
Product Name: DOS Editor
Location: C:\WINNT\system32
Application Type: Word Processing
==========
Product Name: Microsoft 32-bit Registry Editor
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Operating System
Description: Register Programs and Devices
==========
Product Name: Sound Recorder
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Multimedia
Description: Windows 32-bit Sound Recorder
==========
Product Name: Volume Control
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Multimedia
Description: Windows 32-bit Audio Volume
==========
Product Name: Internet Explorer
Vendor Name: Microsoft
Location: C:\Program Files\Plus!\Microsoft Internet
Application Type: Communications
Description: Web Browser

Client Hardware

**********System Information Tool - **********
**********Mouse Information **********
Mouse Type : Microsoft, or IBM PS/2
Number of Buttons : 0
Mouse Sensitivity : 20 mickeys / cm
Vertical Scaling Factor : 10 mickeys / 8 pixels
Horizontal Scaling Factor : 6 mickeys / 8 pixels
Mouse Handedness : Right Handed
Double-click interval : 500 msec
**********Video Subsystem Information **********
Primary Video Adapter reported by Windows
Video Adapter Type : S3 TrioV2 / SGRAM memory
Video Display Type : Unknown Display
Video Memory : 1,024 Kilobytes
Colors Displayed : 65,536
Screen Resolution : 800 x 600
**********Printer Information **********
No Printer Drivers are loaded
**********Memory Information **********
**********Memory address ranges recognized by this system **********
Memory address range under 1 Megabyte
System RAM : 00000000 - 03FFFBFF (65535) Kilobytes
Video RAM : 000A0000 - 000BFFFF (128) Kilobytes

```
Adapter ROMs : 000C0000 - 000DFFFF (128)    Kilobytes
System ROM   : 000E0000 - 000FFFFF (128)    Kilobytes
Memory address range under 16 Megabytes
System RAM   : 00100000 - 00FFFFFF (15,360) Kilobytes
Memory address range above 16 Megabytes
System RAM   : 01000000 - 040FFBFF (50,175) Kilobytes
**********Installed Memory Amounts **********
Memory Amounts by Operating System
Total amount memory detected by Windows : 523,804   Kilobytes
Total amount of virtual memory free     : 1,136,992 Kilobytes
Page size of memory                     : 4,096     Bytes
**********Operating System Information **********
Platform                        : Windows NT Operating System
Windows NT Version              : 5.00
Build Number                    : 2128
Service Pack                    :
Fast Task Switching Enabled     : Yes
USER.EXE Debug Version Installed  : No
Double Byte Character Set Enabled : No
Pen Window Extensions Installed   : No
Total Usable Page File Bytes      : 1248 MB
Available Page File Bytes         : 1110 MB
System Directory                  : C:\WINNT\System32
Windows Directory                 : C:\WINNT
**********Desktop Information **********
Screen Saver Active         : No
Icon Title Wrapping Enabled : Yes
Screen Width                : 800
Screen Height               : 600
**********Window List **********
**********Program Manager **********
Window Handle         : 0001005C
Window Thread ID      : 000002F0
Window Process ID     : 0000028C
Window Visible        : Yes
Window Enabled        : Yes
Window Iconic         : No
Window Maximized      : No
Window Character Type : Unicode
**********Netfinity Applications **********
Window Handle         : 000200EE
Window Thread ID      : 000007A8
Window Process ID     : 00000794
Window Visible        : Yes
Window Enabled        : Yes
Window Iconic         : Yes
Window Maximized      : No
Window Character Type : ANSI-ASCII
**********SYSTEM AGENT COM WINDOW **********
Window Handle         : 0001002C
Window Thread ID      : 00000300
Window Process ID     : 000002C8
Window Visible        : No
Window Enabled        : Yes
Window Iconic         : No
Window Maximized      : No
Window Character Type : Unicode
**********MM Notify Callback **********
Window Handle         : 00020020
Window Thread ID      : 0000034C
Window Process ID     : 000000B4
Window Visible        : No
Window Enabled        : Yes
Window Iconic         : No
Window Maximized      : No
Window Character Type : Unicode
**********00030054 **********
Window Handle         : 00030054
Window Thread ID      : 000002F0
Window Process ID     : 0000028C
Window Visible        : No
Window Enabled        : Yes
Window Iconic         : No
Window Maximized      : No
Window Character Type : Unicode
**********00020052 **********
Window Handle         : 00020052
Window Thread ID      : 000002F0
Window Process ID     : 0000028C
Window Visible        : No
Window Enabled        : Yes
Window Iconic         : No
Window Maximized      : No
Window Character Type : Unicode
**********DDE Server Window **********
Window Handle         : 0002004E
Window Thread ID      : 000002F0
Window Process ID     : 0000028C
Window Visible        : No
Window Enabled        : Yes
Window Iconic         : No
Window Maximized      : No
Window Character Type : Unicode
**********0002004C **********
Window Handle         : 0002004C
Window Thread ID      : 000002F0
Window Process ID     : 0000028C
Window Visible        : No
Window Enabled        : Yes
Window Iconic         : No
Window Maximized      : No
Window Character Type : Unicode
**********MS_WebcheckMonitor **********
Window Handle         : 00010070
Window Thread ID      : 00000420
Window Process ID     : 0000028C
Window Visible        : No
Window Enabled        : Yes
Window Iconic         : No
Window Maximized      : No
Window Character Type : Unicode
**********Connections Tray **********
```

```
Window Handle        : 00010072
Window Thread ID     : 00000434
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********Power Meter **********
Window Handle        : 00010076
Window Thread ID     : 00000448
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********0002009C **********
Window Handle        : 0002009C
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000200A8 **********
Window Handle        : 000200A8
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000100D4 **********
Window Handle        : 000100D4
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000100D6 **********
Window Handle        : 000100D6
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000300DE **********
Window Handle        : 000300DE
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********NETFAPP1 (D:) **********
Window Handle        : 000600B2
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : Yes
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000100E0 **********
Window Handle        : 000100E0
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000100E8 **********
Window Handle        : 000100E8
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000600E4 **********
Window Handle        : 000600E4
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********0003010C **********
Window Handle        : 0003010C
Window Thread ID     : 00000090
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********00010140 **********
Window Handle        : 00010140
Window Thread ID     : 00000090
Window Process ID    : 0000028C
Window Visible       : No
```

```
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********0001013E **********
Window Handle        : 0001013E
Window Thread ID     : 00000090
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********00030108 **********
Window Handle        : 00030108
Window Thread ID     : 00000090
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000300BE **********
Window Handle        : 000300BE
Window Thread ID     : 00000420
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********Alert **********
Window Handle        : 00020168
Window Thread ID     : 00000820
Window Process ID    : 00000824
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********AlertMgr **********
Window Handle        : 00020166
Window Thread ID     : 00000820
Window Process ID    : 00000824
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********Alert Received **********
Window Handle        : 00020164
Window Thread ID     : 00000820
Window Process ID    : 00000824
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000007D8 **********
Window Handle        : 000300C4
Window Thread ID     : 0000027C
Window Process ID    : 000007D8
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********SvcGui **********
Window Handle        : 000300B4
Window Thread ID     : 0000042C
Window Process ID    : 000007D8
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000300B0 **********
Window Handle        : 000300B0
Window Thread ID     : 0000042C
Window Process ID    : 000007D8
Window Visible       : No
Window Enabled       : No
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********Netfinity Help **********
Window Handle        : 00070124
Window Thread ID     : 0000042C
Window Process ID    : 000007D8
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********NetFinityIPC **********
Window Handle        : 00040150
Window Thread ID     : 000007CC
Window Process ID    : 000007D0
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000007D0 **********
Window Handle        : 00040152
Window Thread ID     : 000007C8
Window Process ID    : 000007D0
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
```

```
**********SecMgr **********
Window Handle        : 00040154
Window Thread ID     : 000007D4
Window Process ID    : 000007D0
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********NetMgr **********
Window Handle        : 0003014C
Window Thread ID     : 000007D4
Window Process ID    : 000007D0
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********SvcMgr **********
Window Handle        : 0003014A
Window Thread ID     : 000007D4
Window Process ID    : 000007D0
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000007BC **********
Window Handle        : 00030148
Window Thread ID     : 000007B4
Window Process ID    : 000007BC
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********SCH_BASE_NODE **********
Window Handle        : 00030146
Window Thread ID     : 000007C0
Window Process ID    : 000007BC
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********00030144 **********
Window Handle        : 00030144
Window Thread ID     : 000007C0
Window Process ID    : 000007BC
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000007B0 **********
Window Handle        : 0005010A
Window Thread ID     : 00000814
Window Process ID    : 000007B0
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********PFAServiceBase **********
Window Handle        : 00030142
Window Thread ID     : 000007AC
Window Process ID    : 000007B0
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********00000824 **********
Window Handle        : 0002016A
Window Thread ID     : 00000828
Window Process ID    : 00000824
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********ModemDeviceChange **********
Window Handle        : 0001019A
Window Thread ID     : 00000870
Window Process ID    : 00000848
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000008AC **********
Window Handle        : 000101A2
Window Thread ID     : 000008B0
Window Process ID    : 000008AC
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********ShriekerServiceBase **********
Window Handle        : 000101A4
Window Thread ID     : 000008A8
Window Process ID    : 000008AC
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000101A6 **********
Window Handle        : 000101A6
Window Thread ID     : 000008A8
Window Process ID    : 000008AC
```

```
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000008A4 **********
Window Handle        : 000201A0
Window Thread ID     : 000008A0
Window Process ID    : 000008A4
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********CAPMGT **********
Window Handle        : 0002019E
Window Thread ID     : 0000082C
Window Process ID    : 000008A4
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000008B8 **********
Window Handle        : 0002019C
Window Thread ID     : 000008C0
Window Process ID    : 000008B8
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********MONAOLB **********
Window Handle        : 000101A8
Window Thread ID     : 00000838
Window Process ID    : 000008B8
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********Netfinity **********
Window Handle        : 00030128
Window Thread ID     : 00000090
Window Process ID    : 0000028C
Window Visible       : Yes
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********0005013C **********
Window Handle        : 0005013C
Window Thread ID     : 00000090
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000300FA **********
Window Handle        : 000300FA
Window Thread ID     : 00000090
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********0005014E **********
Window Handle        : 0005014E
Window Thread ID     : 00000090
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********Netfinity Service Manager **********
Window Handle        : 000400AE
Window Thread ID     : 0000042C
Window Process ID    : 000007D8
Window Visible       : Yes
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********0000080C **********
Window Handle        : 000101B4
Window Thread ID     : 0000081C
Window Process ID    : 0000080C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********Gatherer3.0 **********
Window Handle        : 000101B6
Window Thread ID     : 00000810
Window Process ID    : 0000080C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000007B8 **********
Window Handle        : 000101B8
Window Thread ID     : 000007A0
Window Process ID    : 000007B8
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
```

```
Window Character Type : ANSI-ASCII
**********SINF000007B8 **********
Window Handle        : 000101BA
Window Thread ID     : 00000784
Window Process ID    : 000007B8
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********000101BC **********
Window Handle        : 000101BC
Window Thread ID     : 00000784
Window Process ID    : 000007B8
Window Visible       : Yes
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : ANSI-ASCII
**********NetDDE Agent **********
Window Handle        : 0001001E
Window Thread ID     : 00000110
Window Process ID    : 000000B4
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********00030032 **********
Window Handle        : 00030032
Window Thread ID     : 00000420
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********0002004A **********
Window Handle        : 0002004A
Window Thread ID     : 00000420
Window Process ID    : 0000028C
Window Visible       : Yes
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********00030030 **********
Window Handle        : 00030030
Window Thread ID     : 00000420
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********0001006C **********
Window Handle        : 0001006C
Window Thread ID     : 000002F0
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********00020090 **********
Window Handle        : 00020090
Window Thread ID     : 00000420
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********SysFader **********
Window Handle        : 000600C6
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000200A2 **********
Window Handle        : 000200A2
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000700D0 **********
Window Handle        : 000700D0
Window Thread ID     : 000004D8
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********SysFader **********
Window Handle        : 00030116
Window Thread ID     : 00000090
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000400FC **********
Window Handle        : 000400FC
Window Thread ID     : 00000090
```

```
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********0001013A **********
Window Handle        : 0001013A
Window Thread ID     : 00000090
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000F0102 **********
Window Handle        : 000F0102
Window Thread ID     : 00000420
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********SysFader **********
Window Handle        : 00020080
Window Thread ID     : 00000420
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********0001005E **********
Window Handle        : 0001005E
Window Thread ID     : 00000420
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000300B8 **********
Window Handle        : 000300B8
Window Thread ID     : 00000420
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********000300BC **********
Window Handle        : 000300BC
Window Thread ID     : 00000420
Window Process ID    : 0000028C
Window Visible       : No
Window Enabled       : Yes
Window Iconic        : No
Window Maximized     : No
Window Character Type : Unicode
**********Environment Information**********
=C:                  = C:\WNETFIN
ALLUSERSPROFILE      = C:\Documents and Settings\All Users.WINNT
CommonProgramFiles   = C:\Program Files\Common Files
COMPUTERNAME         = SQLCLIENT40
ComSpec              = C:\WINNT\system32\cmd.exe
NUMBER_OF_PROCESSORS = 1
OS                   = Windows_NT
Os2LibPath           = C:\WINNT\system32\os2\dll;
Path                 = C:\WINNT\system32;C:\WINNT;C:\WINNT\System32\Wbem;C:\MSSQL7\BINN
PATHEXT              = .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
PROCESSOR_ARCHITECTURE = x86
PROCESSOR_IDENTIFIER = x86 Family 6 Model 7 Stepping 3, GenuineIntel
PROCESSOR_LEVEL      = 6
PROCESSOR_REVISION   = 0703
ProgramFiles         = C:\Program Files
SystemDrive          = C:
SystemRoot           = C:\WINNT
TEMP                 = C:\WINNT\TEMP
TMP                  = C:\WINNT\TEMP
USERPROFILE          = C:\Documents and Settings\Default User.WINNT
windir               = C:\WINNT
**********Model and Processor Information **********
Processor            : Unknown
Math Coprocessor     : Not Installed
Number of Processors : 00
Expansion Bus Type   : ISA or AT bus
Model                : 00
Submodel             : 00
BIOS Revision        : 00
**********Keyboard Information **********
Keyboard Type   : 101/102 Key Enhanced Keyboard
Code Page       : 437
Typematic Rate  : 31 characters/sec
Typematic Delay : 1 msec
**********Security Information **********
Security features present:
  Power on password feature available
  Privileged Access password feature available
  Unattended start mode feature available
  Machine tamper detection feature available
  Machine has lockable diskette drives
**********SCSI Subsystem Information **********
**********SCSI Adapter Information **********

Bus Type             : SCSI 2
Location             : System Board Controller
Bus Attributes       : Not Available
I/O Access           : IO Message - Other
Host Bus             : Host Bus - other
Host Bus Width       : Not Available
Adapter Attributes   : Supports addresses below 16 MB
```

```
Adapter Attributes  : Does not support IBM SCB commands
Adapter Attributes  : Supports scatter/gather in software
Adapter Attributes  : Supports CHS addressing in hardware
Maximum CDB Length  : Not Applicable
Device(s) connected : 0
**********SCSI Adapter Information **********

Bus Type          : SCSI 2
Location          : System Board Controller
Bus Attributes    : Not Available
I/O Access        : IO Message - Other
Host Bus          : Host Bus - other
Host Bus Width    : Not Available
Adapter Attributes  : Supports addresses below 16 MB
Adapter Attributes  : Does not support IBM SCB commands
Adapter Attributes  : Supports scatter/gather in software
Adapter Attributes  : Supports CHS addressing in hardware
Maximum CDB Length  : Not Applicable
Device(s) connected : 2
Fixed Disk
Unknown
**********SCSI Devices For Adapter **********
Device Type          : Fixed Disk
Device Size          : 4,406,935 Kilobytes
Device Unit PUN      : 0
Device Unit LUN      : 0
ANSI Level Supported  :
Unit Status          : Ready and Powered On
Vendor ID            : IBM
Product ID           : DCHS04Y
Product Revision Level : 6363
Vendor Field         : 6819B333RAMR6063
Device Attributes    : Device supports 16-bit wide data transfers
Device Attributes    : Device supports synchronous data transfers
Device Attributes    : Device supports linked commands
Device Attributes    : Device supports tagged command queuing
Device Attributes    : Prefetch Not Supported
Device Attributes    : Device Not Defective
**********SCSI Devices For Adapter **********
Device Type        : Unknown
Device Size        : Not Available
Device Unit PUN    : 14
Device Unit LUN    : 0
ANSI Level Supported : May not comply to ANSI SCSI
Unit Status        : Powered On
Device Attributes  : Prefetch Not Supported
Device Attributes  : Device Not Defective
**********IDE Subsystem Information**********
**********IDE Adapter Information **********

Bus Type          : ST-506 CAM-I
Location          : System Board Controller
Bus Attributes    : Not Available
I/O Access        : IO Message - Other
Host Bus          : Host Bus - other
Host Bus Width    : Not Available
Adapter Attributes  : Supports addresses below 16 MB
Adapter Attributes  : Does not support IBM SCB commands
Adapter Attributes  : Supports scatter/gather in software
Adapter Attributes  : Supports CHS addressing in hardware
Maximum CDB Length  : Not Applicable
Device(s) connected : 1
CD ROM Device
**********ST506/IDE Devices For Adapter **********
Device Type      : CD ROM Device
Device Size      : 512,200 Kilobytes
Unit Status      : Ready and Powered On
Media Status     : Present
Product ID       : SANYO CD-ROM CRD-1332P
Device Attributes : Prefetch Not Supported
Device Attributes : Device Not Defective
**********IDE Adapter Information **********

Bus Type          : ST-506 CAM-I
Location          : System Board Controller
Bus Attributes    : Not Available
I/O Access        : IO Message - Other
Host Bus          : Host Bus - other
Host Bus Width    : Not Available
Adapter Attributes  : Supports addresses below 16 MB
Adapter Attributes  : Does not support IBM SCB commands
Adapter Attributes  : Supports scatter/gather in software
Adapter Attributes  : Supports CHS addressing in hardware
Maximum CDB Length  : Not Applicable
Device(s) connected : 0
**********System Information Tool -  **********
```

Client Software

```
System Name:
Dictionary Name: C:\WNETFIN\default.sid
============================================================
Product Name: Notepad
Vendor Name: Microsoft
Location: C:\WINNT
Application Type: Word Processing
Description: Windows Notepad Editor
============================================================
Product Name: Netfinity Services for Windows (32 bit) v. 5.20.x
Vendor Name: IBM
Version: 5.20
Revision: x
Location: C:\WNETFIN
Application Type: System Management
Description: Network Management Suite
============================================================
Product Name: Netfinity Manager for Windows (32 bit) v. 5.20.x
Vendor Name: IBM
Version: 5.20
Revision: x
```

Location: C:\WNETFIN
Application Type: System Management
Description: Network Management Suite
==========
Product Name: Internet Explorer
Vendor Name: Microsoft
Location: C:\Program Files\Internet Explorer
Application Type: Communications
Description: Web Browser
==========
Product Name: Phone Dialer
Vendor Name: Microsoft
Location: C:\Program Files\Windows NT
Application Type: Communications
Description: Windows Telephone Dialer
==========
Product Name: CD Player
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Multimedia
Description: Windows CD Player
==========
Product Name: Character Map
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Word Processing
Description: Windows Character Map
==========
Product Name: Clipboard Viewer
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Word Processing
Description: Windows Clipboard Viewer
==========
Product Name: Freecell
Vendor Name: Microsoft Corp.
Location: C:\WINNT\system32
Application Type: Entertainment
==========
Product Name: Notepad
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Word Processing
Description: Windows Notepad Editor
==========
Product Name: Microsoft 32-bit Registry Editor
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Operating System
Description: Register Programs and Devices
==========
Product Name: Sound Recorder
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Multimedia
Description: Windows 32-bit Sound Recorder
==========
Product Name: Volume Control
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Multimedia
Description: Windows 32-bit Audio Volume
==========
Product Name: Solitaire
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Entertainment
Description: Windows Card Game
==========
Product Name: Minesweeper
Vendor Name: Microsoft
Location: C:\WINNT\system32
Application Type: Entertainment
Description: Windows Game
==========
Product Name: Pinball
Vendor Name: Maxis
Location: C:\Program Files\Windows NT\Pinball
Application Type: Entertainment
Description: Desktop Pinball Simulator
==========
Product Name: CD Player
Vendor Name: Microsoft
Location: C:\WINNT\system32\dllcache
Application Type: Multimedia
Description: Windows CD Player
==========
Product Name: Character Map
Vendor Name: Microsoft
Location: C:\WINNT\system32\dllcache
Application Type: Word Processing
Description: Windows Character Map
==========
Product Name: Clipboard Viewer
Vendor Name: Microsoft
Location: C:\WINNT\system32\dllcache
Application Type: Word Processing
Description: Windows Clipboard Viewer
==========
Product Name: Phone Dialer
Vendor Name: Microsoft
Location: C:\WINNT\system32\dllcache
Application Type: Communications
Description: Windows Telephone Dialer
==========
Product Name: Freecell
Vendor Name: Microsoft Corp.
Location: C:\WINNT\system32\dllcache
Application Type: Entertainment
==========
Product Name: Internet Explorer
Vendor Name: Microsoft

Location: C:\WINNT\system32\dllcache
Application Type: Communications
Description: Web Browser
==========

Product Name: Notepad
Vendor Name: Microsoft
Location: C:\WINNT\system32\dllcache
Application Type: Word Processing
Description: Windows Notepad Editor
==========

Product Name: Microsoft 32-bit Registry Editor
Vendor Name: Microsoft
Location: C:\WINNT\system32\dllcache
Application Type: Operating System
Description: Register Programs and Devices
==========

Product Name: Sound Recorder
Vendor Name: Microsoft
Location: C:\WINNT\system32\dllcache
Application Type: Multimedia
Description: Windows 32-bit Sound Recorder
==========

Product Name: Volume Control
Vendor Name: Microsoft
Location: C:\WINNT\system32\dllcache
Application Type: Multimedia
Description: Windows 32-bit Audio Volume
==========

Product Name: Solitaire
Vendor Name: Microsoft
Location: C:\WINNT\system32\dllcache
Application Type: Entertainment
Description: Windows Card Game
==========

Product Name: Minesweeper
Vendor Name: Microsoft
Location: C:\WINNT\system32\dllcache
Application Type: Entertainment
Description: Windows Game

Appendix E: 180-Day Space

TPC-C 180-Day Space Requirements						
Warehouses	3,254				tpmC	40,251.15
Table	Rows	Data KB	Index KB	Extra 5% KB	8HR Space	Total Space KB
Warehouse	3,254	352	64	21		437
District	32,540	3,616	88	185		3,889
Item	100,000	9,528	112	482		10,122
New-Order	29,286,000	463,024	1,264		260,320	724,608
History	97,620,000	5,423,504	0		1,073,398	6,496,902
Orders	97,620,000	2,992,184	1,652,704		919,297	5,564,185
Customer	97,620,000	70,996,368	4,558,848	3,777,761		79,332,977
Order-Line	976,202,542	60,012,664	151,944		12,105,450	73,270,058
Stock	325,400,000	104,128,000	233,360	5,218,068		109,579,428
Total		245,029,240	6,598,384	8,996,517	14,358,465	274,982,605
Segment	LogDev Cnt.	Segment Size	Needed	Overhead		Not Needed
misc	10	98,304,000	86,070,201	860,702		11,373,097
cs	10	196,608,000	188,912,405	1,889,124		5,806,471
master, msdb,model	1	13,312	13,312			0
tpcc_root	1	8,192	8,192			0
tempdb	1	8,704	8,704			0
Totals		294,942,208	275,012,813	2,749,826		17,179,568
Dynamic Space	69,428,352	Sum of Data for Order, Order-Line and History				
Static Space	193,945,615	Data + Index + 5% Space + Overhead - Dynamic Space				
Free Space	14,388,673	Total Segment Size - Dynamic Space - Static Space - Not needed				
Daily Growth	13,740,976	(Dynamic Space/W * 62.5)* tpmC				
Daily Spread	(6,222,791)	Free Space - 1.5 * Daily Growth (Zero If Negative)				
180-Day Space (KB)	2,667,321,310	Static Space + 180 (Daily Growth + Daily Spread)				
180-Day Space (GB)	2,543.76	180-Day Space in GB (Excludes OS,Paging and RDBMS Logs)				
Log size (MB)	60,000	Total Size of Log File				
% log used	15.0589	% of Log File Used During Entire Run				
Total N-O Txn	1,846,673	Total Count of New-Order Transactions during Entire Run				
Log / N-O Txn	5.0102	KB of Log per New-Order Transaction				
8 Hour log (GB)	92.32	8 Hours of Log in GB (Excluding Space for Redundancy)				
Disk Capacity	MB	GB				
9.1GB	8,676	8.47				
18.2GB	17,318	16.91				
Space Usage	GB Needed	Disks Measured	Disks Priced	Disk Size	Formatted Size	
180-Day Space DB	2,543.76	400	400	9.1GB	3,389.45	
Extra Disks		0	0	9.1GB	0	
Total DB	2,543.76	400	400		3,389.45	
8hr Log + Mirror	92.32	12	12	18.2GB	103.91	
OS, SQL Server	4.00	1	1	9.1GB	8.47	
Total Space	2640.08	413	413		3,501.83	

Appendix F: Third-Party Quotations

Software House Internati

Pricing Proposal

Matthew Martin: National Account Executive

Phone: 408-922-1106 Fax: 408-526-1222

Description	Part Number	Unit Price	Qty	Extended Price
Server Hardware				
IBM Netfinity 8500R	8681-6RY	23,330	1	$23,330
550MHz / 2MB Processor Upgrade	33L5105	6,280	7	43,960
8500 Memory Expansion Card	28L4454	587	1	587
8500 >4-way Enablement Kit (4X SRAM)	28L4727	1,990	1	1,990
512MB ECC SDRAM RDIMM Memory Upgrade Kit	20L0249	2,275	7	15,925
Netfinity 4.2M Ultra2 SCSI Cable	03K9311	121	44	5,324
Netfinity Fibre Channel RAID Controller Unit	35261RU	12,560	1	12,560
Netfinity PCI Fibre Channel Adapter	01K7297	1,675	1	1,675
Netfinity Fibre Channel 5M Short-Wave Cable	03K9306	127	1	127
EtherJet 10/100 PCI Management Adapter	34L1210	58	1	58
IBM G42 14" (13.2" Viewable) Color Monitor	654000N	169	1	169
20/40GB DLT SCSI Internal Tape Drive	01K1320	1,955	1	1,955
Netfinity Rack	9306900	1,685	4	6,740
Side Panel Kit	94G6669	197	1	197
Server Storage				
IBM Netfinity EXP200 Rack Storage Exp. Enclosure	35301RU	2,510	42	105,420
Netfinity 9.1GB 10K Wide Ultra SCSI Drive	36L9748	469	401	188,069
Netfinity 18.2GB 10K Wide Ultra SCSI Drive	36L9749	837	12	10,044
Client Hardware				
IBM Netfinity 5000 / 600MHz/512KB Pentium III	8659-6RY	3,035	6	$18,210
128MB 100MHz ECC SDRAM RDIMM	01K7262	417	18	7,506
9.1GB 10K rpm Wide Ultra Drive	36L9806	471	6	2,826
Intel Pro/100+ Dual Port Ethernet Adapter		69	12	828
Netfinity 10/100 Ethernet PCI Adapter	34L1501	101	12	1,212
IBM G42 14" (13.2" Viewable) Color Monitor	654000N	169	6	1,014
8port + 1 BNC 10BT Hub	DEH2924	25	4,475	111,875
Total				$561,601

5 year return to man warranty on hubs.
Quote good for ninety days.
18-Feb-00

One Microsoft Way
Redmond, WA 98052-6399
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

February 22, 2000

IBM Corporation
Chris King
kchris@us.ibm.com
919-543-8799
919-254-9267 fax

Dear Mr. King:

Here is the information you requested regarding U.S. pricing of several Microsoft products that were used in a recent TPC-C benchmark:

Microsoft SQL Server 7.0, Enterprise Edition (one server plus unlimited CALs)	**$28,999**
Microsoft Windows NT Server 4.0, Enterprise Edition (one server plus 25 CALs)	**$3,999**
Microsoft Windows 2000 Server 4.0 (one server with 25 CALs)	**$809**
Visual C++ Professional 5.0 (single copy)	**$549**
5-year maintenance for above software @ $2095/yr	**$10,475**

This quote is valid for the next 90 days. Some products may not be currently orderable but will be available through Microsoft's normal distribution by July 31, 2000.

If I can be of any further assistance, please contact me at 425-936-5301 or tomkr@microsoft.com.

Yours truly,

Thomas Kreyche
Product Manager
SQL Server Marketing

American Megatrends

Date:	02/23/00	**e-mail**	kchris@us.ibm.com
Company:	IBM	**Phone Number:**	(919) 543/ 8-441-8799
Contact Name:	Chris	**Fax Number:**	

MegaRAID Series Enterprise 1500-H PCI-SCSI Raid Controller

Item	Part No.	Qty 1-10
MegaRAID Series Enterprise 1500-H High Performance RAID Controller, 4 Channel, 64MB SDRAM Memory, NT Monolithic Driver, Firmware, Battery Back, and Manuals	4383536116	$1750.00
Extended 2 year limited warranty		$105.00 Per Unit

MegaRaid Enterprise 1500-H Features

- Intel i960RN Processor @ 66MHz
- 64 bit PCI
- LVD or Single Ended Support
- On board Battery Circuitry
- 4 Internal and 4 External Connectors
- Supports Ultra 2 Speeds of 80/mb sec
- High Performance Firmware

Deliverables

1. Windows NT Monolitic Driver
2. Utilities: Control M Boot up Configuration.
3. Manuals (Optional): Hardware Guide, Software Guide and Driver Installation Guide. (Hard and/or Soft Copy available)

Conditions:

1. All pricing is quoted FOB factory, Norcross, GA; shipping and insurance are additional.
2. The Lead-time on forecasted orders (6 months forecast requested) is four weeks ARO.
3. Three-year limited warranty on parts and labor is offered for hardware products.
4. The Users Manual and the software drivers are included, however American Megatrends retains all copyrights.
5. Payment terms must be established with American Megatrends prior to any shipments.
6. Pricing will be adjusted should memory fluctuate +/- 5%.

Submitted by: Elaine Morris **Date: February 16, 2000**

This quotation is valid for 90 days from the date shown and is subject to the conditions as listed.

NETLUX

14180 Live Oak Ave., Unit E
Baldwin Park, Ca. 91760

1-800-789-1780
Phone#626-851-9737
Fax #626-851-9837

February 23, 2000

Chris King
IBM Corp.
(919) 254-9267

Quotation

Quantity	Part No.	Description	Unit Price	Total
3	NX-DS8	8-port 10/100Mbps FAST Ethernet Hub	$95.00	$ 285.00

Terms and Conditions:
FOB Origin
Quote Valid for 90 days
5 Year Warranty

Sincerely,
Martin Parry
NETLUX